12

NRL Memorandum Report 3082

A Fortran Program to Plot Phase Shifts of the Three-Dimensional Square Well

James L. Remack

Solid State Devices Branch
Electronics Technology Division

July 1975

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 3082	2. GOVT ACCESSION NO. (9) Final rept.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A FORTRAN PROGRAM TO PLOT PHASE SHIFTS OF THE THREE DIMENSIONAL SQUARE WELL.		5. TYPE OF REPORT & PERIOD COVERED A final report on one phase of a continuing NRL problem.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) James L. Repace		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) NRL ~~Problem~~ R08-42 ~~NRL Thomas Edison Program~~
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE (11) Jul[redacted]75
		13. NUMBER OF PAGES 46
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 47p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(14) NRL-MR-3082

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

3 dimensional square well
Potential scattering
Phase shifts

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This program calculates, and provides CalComp plots of, the CDC 3800 Computer, phase shifts in radians or degrees for quantum-mechanical scattering from a 3-dimensional spherically symmetrical square well potential of width a and depth −V, for a de Broglie quantum particle of energy E. The program, which can plot in two or three dimensional representations, calculates the first five phase shifts, and permits the determination of the number of bound states the well will support for each phase shift.

251 950

mb

...URITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

A FORTRAN PROGRAM TO PLOT PHASE SHIFTS OF THE THREE DIMENSIONAL SQUARE WELL

INTRODUCTION

This program calculates, and provides CalComp plots of, phase shifts in radians or degrees for quantum mechanical scattering from a 3-dimensional spherically symmetrical square well potential of width a and depth -V, for a de Broglie quantum particle of energy E. It exists in two versions, one of which provides plots of the phase shift δ_ℓ vs Ea^2, with the phase shift given in units of pi and Ea^2 in dimensionless units, in a standard two dimensional CalComp plot. The other version adds a third dimension to the plot, with δ_ℓ being plotted vs Ea^2 and V. The program permits the first five phase shifts to be calculated (ℓ values 0 thru 4) and permits a determination of the number of s-wave bound states the well will support via Levinson's Theorem.

1.0 IDENTIFICATION

1.1 Title

Calculation and plotting routine for determination of phase shifts of the spherically symmetrical square well potercial.

1.2 Entry points: Both versions

PHASHIFT (2D and 3D versions)

SPHJN

Note Manuscript submitted June 23, 1978

SPHYN

MINIMAX

LAYOUT (2D and 3D versions)

1.3 Programming Language

Language: 3600/3800 FORTRAN

Routine Type: Program, Subroutine

Operating System: DRUM SCOPE 2.1

1.4 Computer and Configuration

CDC 3800

1.5 Contributor or Programmer

J. L. Repace, Code 5216J, Solid State Devices Branch, Electronics Technology Division

1.6 Contributing Organization

NRL - Naval Research Laboratory, Washington, D.C. 20375

1.7 Program Availability

1.7.1 Submittal Program writeup, FORTRAN source deck, source listing

1.8 This program has been used to calculate and plot in two and three dimensional perspective the phase shifts of the spherically symmetric square well potential for square wells of depth 0 thru 1000 in dimensionless units, for energies 0 thru 5000 in dimensionless units, for s thru g-wave scattering. (VERIFICATION)

1.9 1 June 1975

2.0 PURPOSE

Although the general properties of the phase shift of the three dimensional spherically-symmetrical square well potential* are described in most textbooks on quantum mechanics, specific graphical representations of this function are rare.

*Shown in Fig. 1

This program permits two and three dimensional Calcomp plots of s, p, d, f and g-wave phase shifts, as a function of energy.

Input is handled by PHASHIFT, with the well parameters energy limits and plotting information being read from data cards. The phashifts are then calculated, with PHASHIFT calling SPHYN and SPHJN, routines which respectively compute spherical Neumann and spherical Bessel functions. Then PHASHIFT calls MINIMAX, which computes the largest and smallest phashifts calculated if the plots are to be automatically scaled. Finally, the plotting routine, LAYOUT is called and a Calcomp plot is produced.

2.1 Problem Background

In many fields of physics, observations are described in terms of the scattering of an electron, or other particle, on a target represented by a three dimensional square well. Examples are found in nuclear physics, in atomic physics, and in solid state theory. Often, the square well approximation is only used initially to obtain some idea about the size of the scattere and of the strength of its potential. In many cases the approximation is augmented with a variety of refinements. Nevertheless, the three dimensional square well provides us with the prototype of the behaviour of a scattering object. This is the case in quantum mechanics, and also in electromagnetic theory. Although the general properties of the phase shift of the three dimensional spherically-symmetrical square well potential are described in most textbooks on quantum mechanics, specific graphical representations of this function are rare.

3.0 USAGE

3.1 Calling Sequence or Operational Procedure

To use PHASHIFT as a program, data cards after the $LOAD card are required. See section 3.7. There are two different versions, depending upon whether two dimensional plot displays one through five phase shifts versus the energy-well-width parameter Ea^2. The three dimensional plot displays a selected phase shift versus Ea^2 for up to nine different potential depths, where the potential depths are displayed on the third axis, which is oriented at an angle of about 20° from the abscissa. The well depths are displayed equally spaced along the potential axis, although they need not be consecutive.

3.2 Arguments, Parameters, and Initial Conditions

2-D Version and 3-D Version

ZERO - Lowest energy value

EPSILON - The increment to ZERO for each iteration, of which there are LAST

LAST - The total number of data points desired (EPSILON X LAST = the maximum value of the energy)

VULCAN - the well depth

FERMI - the well width

MAX - The highest order phase shift desired (all phase shifts up to MAX are superimposed on one plot)

HELLO - determines whether the phase shifts are given in degrees or radians

NOPLOT suppresses plotting

NOSCALE suppresses ordinal self-scaling: if self-scaling is suppressed, maximum and minimum values of the phase shift YMAX, YMIN must be inserted prior to statement 100 in PHASHIFT

KILL Designated last data card: the position of KILL on the data cards differs between the 2-D and 3-D versions.

For additional information concerning these parameters, see Section 3.7.

3.3 Space Required (Decimal and Octal):

3.3.1 Unique Storage:

2-D version:

PHASHIFT	12722	octal	Locations
MINIMAX	00230	"	"
SPHJN	01140	"	"
SPHYN	00277	"	"
LAYOUT	01332	"	"
SCALE	00246	"	"
AXIS	00536	"	"

3-D version:

PHASHIFT	13003	octal	Locations
MINIMAX	00230	"	"
SPHJN	01140	"	"
SPHYN	00277	"	"
LAYOUT	01332	"	"
SCALE	00246	"	"
AXIS	00536	"	"

3.3.2 Common Blocks:

Both Versions:

```
COMMON/LABEL/IL, ILX(5), ILY(5),ILH(5),ILR(5),ILC(5),ILABEL(5,5)
COMMON/RANGE/XMAX,XMIN,YMAX,YMIN,IFRST(5),ILST(5),XSIZE
(In the 3-D version the subscripts for all variables and arrays in LABEL and RANGE
are (9) and (9,9) resp.)
COMMON/1/DELTA(1000),CASTOR(1000),POLLUX(1000),HERCULES(1000),AJA
1X(1000),OLYMPUS(1000),HADES(1000),B(1000),DB(1000),N(1000),DN(1000
2),BZ(1000),DBZ(1000),NZ(1000),GAMMA(1000),ENERGY(1000),CYCLOPS(100
30),CENTAUR(1000),ZEUS(1000),RHO(1000),DNZ(1000)
COMMON/A/PHASE(5000),ERGON(5000) (In 3-D version this is called COMMON/3/)
```

3.3.3 Temporary Storage: None

3.4 Messages and Instructions to Operator: None

3.5 Error returns, Messages and Codes

These are all related to the Bessel and Neumann function subroutines: See NRL Computer Bulletin 29, section 3.6 for explanation of error messages relating to overflow and underflow prevention and use of zero arg. for Neumann functions.

3.6 Informative Messages to the User

Other than the error messages mentioned in Section 3.5, no informative messages are printed.

3.7 Input

This section describes the data required for both the 3-D and 2-D versions of Program Phashift.

3.7.1 2-D Version: One card is required to control the input of data. This card specifies:

Parameter	Format	Columns	Description
EZERO	F10.5	1-10	The smallest energy value desired in the calculation of the phase-shift vs energy plot.
EPSILON	F10.5	11-20	The energy increment added to EZERO to generate the next energy data point.
VULCAN	F10.5	21-30	The absolute value of the well depth.
FERMI	F10.5	31-40	The well width.
MAX	I1	45	The highest order phase shift desired. For S,P,D,F,G-waves, MAX = 1,2,3,4,5.
JELLO	I1	50	Controls units of phase shift. If JELLO is 1, phase shift is expressed in degrees, if JELLO is blank, phase shift is calculated in radians.
LAST	I4	55-58	The total number of data points desired to be calculated, maximum 1000. The largest energy value calculated is given by the product EPSILON X LAST.
NOPLOT	I1	65	If a plot is not desired, enter a 1. Otherwise leave blank.
NOSCALE	I1	70	If set = 1, ordinal scale is 0-1. If blank, automatic self-scaling obtains.
KILL	I1	72	Zero except in last data card of set

3.7.2 3-D Version: Two-plus cards are required to control the input of data. A maximum of 10 data cards per set are possible. The first card specifies:

Parameter	Format	Columns	Description
EZERO	F10.7	1-10	The smallest energy value desired in the calculation of the phase-shift vs energy plot.
EPSILON	F10.7	11-10	The energy increment added to EZERO to generate the next energy data point.
FERMI	F10.7	21-30	The well width.
MAX	I1	45	The highest order phase shift desired. For S,P,D,F,G-waves, MAX = 1,2,3,4,5.
JELLO	I1	50	Controls units of phase shift. If set to 1, units are degrees, if blank, units are radians.
LAST	I4	55-58	The total number of data points desired to be calculated, maxmum 1000. The largest energy value calculated is given by product EPSILON X LAST.
NCURVES	I1	60	The number of curves of different well depth desired to be plotted (maximum of 9).
NOPLOT	I1	65	If a plot is not desired, enter a 1, otherwise leave blank.
NOSCALE	I1	70	If set 1, ordinal scale is 0-1. If blank, automatic self-scaling obtains.

The second and remaining cards specify:

VULCAN	F10.5	1-10	The absolute value of the well depth. Successive data cards generally contain successively larger well depths, up to a maximum of 9 total data cards giving values for VULCAN.
KILL	I1	21	Zero, except on the last data card of each successive data set.

3.8 Output

3.8.1 Printed output: Both versions

The following information is printed before the data list:

(1) PHASE SHIFT IN RADIANS (The word DEGREES will appear instead of RADIANS if that option is selected) FOR A QUANTUM MECHANICAL PARTICLE SCATTERING OFF A 3-D SQUARE WELL.

(2) Information regarding the chosen input parameters is printed as follows:

WELL DEPTH =
WELL WIDTH =
EZERO = } The minimum and maximum values along the energy axis.
ELAST =

(3) Data List: The data list contains 10 variables as follows:

ENERGY (I): The energy of the Ith data point

DELTA (0): The zero order phase shift (S-wave)

DELTA (1): The first order phase shift (P-wave)

DELTA (2): The second order phase shift (D-wave)

DELTA (3): The third order phase shift (F-wave)

DELTA (4): The fourth order phase shift (G-wave)

K : ZEUS(I), the product of the square root of the ENERGY(I) and FERMI (the well width)

ERGO : The product K x FERMI (this is what the actual abscissal value is in the plots)

OLYMPUS 1: The numerator of the Arctan function for I=1 data point (printed for diagnostic purposes in checking the accuracy of the phase shift calculation) see sect. 4.0.

HADES 1 : The denominator of the Arctan function described above.

3.8.2 Plotted Output

2-D version: Samples of the plotted output are given in figures 4, 5, and 6. Each of these figures shows the S, P, D, F, and G-wave phase shifts superimposed on the same plot for comparative purposes. The ordinal values give the phase shifts in units of pi, while the abscissal values are products of energy

× well width^2. The curves shown have been embellished for illustrative purposes, the actual curves are all displayed as solid lines, and must be identified individually either by plotting consecutive displays working up from S to G-waves, or by recourse to the printed output. The interpretation of the broken curves displayed in figure 6 is discussed in section 8.0 (REMARKS); it is a result of the indistinguishability of the multiples of $n\pi$ to the computer.

3-D version: A sample of the plotted output is given in figure 3. It is embellished only by the numerals identifying V-values near each curve, these are not displayed on the CalComp plot, otherwise the figure is essentially unretouched. It should be noted that it is not necessary to have sequential V-values; it should also be noted that the maximum point of each individual V curve is displayed between 0 and 1 in units of pi, and in no case is greater than 1. In cases where the curve would have penetrated the region between 1 and 2, the computer displays the peak of the curve reduced to the interval 0 to 1, as in curve V = 20.

3.9 Formats

In the 2-D version, the data are printed in F10.5 format; in the 3-D version, the data are printed in F11.7 format.

3.10 External Routines and Symbols

```
Both versions:     PHASHIFT calls     SPHJN    ⎫
                                      SPHYN    ⎪
                                      MINIMAX  ⎬      -PHASHFT
                                      LAYOUT   ⎪
                                      STOPPLOT ⎭

                                      ATAN2    ⎫
                                      SQRTF    ⎭      -SYSTEM LIBRARY
```

3.11 Timing

2-D version: To produce figures such as figures 4, 5 and 6, i.e., for 1000 data points and calculations of S thru G wave phase shifts, the processing time is generally less than 1.5 minutes. The plotting time generally runs about 4 minutes.

3-D Version: To produce a figure such as figure 3 generally takes about 4 minutes of computer time plus about 4 minutes of CalComp time.

3.12 Accuracy

The accuracy of the calculations was verified using an electronic calculator for several points on the S-wave curves. The accuracy of the plots was verified by comparison with curves for S, P, and D-wave phase shifts presented for V = 57.5 over an Ea^2 interval of 0 to 10, which were presented in Mott and Massey[1].

3.13 Cautions to Users

Experience has shown that the main sources of error in using these routines lies in the confusion which may arise in identifying tne individual phase shift curves on the Calcomp plot, especially at the high energy end of the curves.

A second source of confusion may arise from the display of broken curves in figures 3 and 6 due to the compression by the computer of multiples of pi into the first "zone" of zero to pi. This reduction is demonstrated in figures 2a and 2b, the unreduced and reduced phase shifts as applied to figure 5a, page 33, in Mott and Massey. A method of determining the ordinal intercept, so as to be able to apply Levinson's theorem, which relates the ordinal intercept to the number of bound states the well may support, is discussed in section 8.o, REMARKS.

3.14 Program Deck Structure

2-D version: $^7_9$JOB card

$^7_9$EQUIP, 13 = PL

$^7_9$FTN card

Program Subroutines: PHASHIFT
SPHJN
SPHYN
MINIMAX
LAYOUT (with COMMON blocks and DIMENSION statements
SCOPE subscripted 5)

$^7_9$LOAD card

$^{7}_{9}$RUN card

data cards (see sect. 3.7.1) each data card will generate a series of superimposed phase shift curves for a given well depth; each successive data card will generate another plot.

EØF

3-D version: $^{7}_{9}$JOB card

$^{7}_{9}$DEMAND, 53776B

$^{7}_{9}$EQUIP, 13 = PL

$^{7}_{9}$FTN

```
Program subroutines:  PHASHIFT
                      SPHJN
                      SPHYN
                      MINIMAX
                      LAYOUT  (with COMMON blocks and DIMENSION statements
                      SCOPE    subscripted 9)
```

$^{7}_{9}$BANK, (0',/3/)

$^{7}_{9}$LOAD card

$^{7}_{9}$RUN card

data cards (see section 3.7.2)

EØF

3.15 References - Literature - Appendices

(1) Mott, N. F. & Massey, H. S. W., THE THEORY OF ATOMIC COLLISIONS, 3rd Ed., Oxford, Clarendon Press, 1965.

(2) Meijer, P. H. E. & Repace, J. L., "Phase Shifts of the 3-Dimensional Spherically Symmetric Square Well Potential", American Journal of Physics, Vol. 43, No. 5, May 1975, p. 428.

(3) Newton, R. G., SCATTERING THEORY OF WAVES AND PARTICLES, McGraw-Hill, N.Y., 1968.

4.0 METHOD OR ALGORITHM

I. A detailed description of the significance of certain features of the phase shift curves, as well as a guide to the literature is available in reference

(2). The mathematical expression for the phase shifts and simplifying assumptions regarding the constants are discussed below.

4.1 Description of the Phase Shift Equations

We now consider the ideas of the partial wave method. Suppose we have a solution inside a spherical well of radius a and depth V. See figure 1. This solution has to match the wavefunctions outside the well at r = a. At this point the wavefunction should have the same value (continuity of the wavefunction) and the same derivative (continuity of the derivative of the wave function). The continuity of the derivative of the wavefunction is required because the momentum of the particle should be defined at every point in space. There are two more boundary conditions: 1) the wavefunction inside the sphere should be zero at the origin, and 2) the wavefunction outside the sphere should have a fixed amplitude. The value of this amplitude is of no importance when we calculate the phase shift. The first boundary condition excludes Neumann functions inside the sphere. As a result of the spherical symmetry of the potential well, the angular dependence inside and outside should be identical: the ℓ-value outside the sphere.

We introduce the following notation. $k = \sqrt{E}$. We use units $m = \hbar = 1$ and $\nu = \sqrt{(V+E)}$ where V = the well depth, the absolute value of the negative potential inside the sphere. We use the standard notation for the spherical Bessel and Neumann functions (compare NBS handbook)[10]. The radius of the sphere is $r = a$. With this notation we have, for the radial part of the wavefunction inside the sphere:

$$r<a: \; \psi(r) = Aj_\ell(\nu r)$$

and for the wavefunction outside the sphere:

$$r>a: \; \psi(r) = Bj_\ell(kr) + Cn_\ell(kr), \; B^2+C^2 = 1$$

The convention is normally to replace the two constants by the sine and cosine of an angle as follows.

Having established there two wavefunctions, the matching condition gives the following relation:

$$j_\ell(\alpha a) = \cos\delta_\ell j_\ell(ka) - \sin\delta_\ell n_\ell(ka)$$

$$\alpha j'_\ell(\alpha a) = k\ (\cos\delta_\ell j'_\ell(ka) - \sin\delta_\ell n'_\ell(ka))$$

We take the ratio between these two equations, omitting the subscript

$$\alpha\frac{j'(\alpha a)}{j(\alpha a)} = \frac{kj'(ka) - k\tan\delta\ n'(ka)}{j(ka)\quad - \tan\delta\ n(ka)}$$

and solve for tan δ. The result can be written as

$$\delta = \tan\text{-}1\left[\frac{ka\ j'(ka)\ -\ \gamma\ j(ka)}{ka\ n'(ka)\ -\ \gamma\ n(ka)}\right]$$

where γ is given by

$$\gamma = \alpha a\ j'(\alpha a)/j(\alpha a)$$

These are the equations of interest which are used to compute the phase shift δ_ℓ.

5.0 Listing of program — see pages 23 et. seq.

6.0 COMPARISONS - see section 3.12

7.0 TEST METHOD AND RESULTS

2-D version: To generate figures 4, 5 and 6 during one computer run, set

1st data card	2nd data card	3rd data card
EZERO .00001	.00001	.00001
EPSILON = 0.5	0.5	0.5
VULCAN = 1.0	10.0	100.0
FERMI = 1.0	1.0	1.0
MAX = 5	5	5
JELLO = 6	6	6
LAST = 100	100	100
NOPLOT = 6	6	6
NOSCALE = 1	1	1
KILL = 0	0	1

The partial printed output from the second data card follows: (See Figure A)

3-D version: To generate figure 3, set

```
     1st data card

     EZERO            .0000001
     EPSILON          .014
     FERMI            1.
     MAX              5
     NFLLO            b
     LAST             1000
     NCURVES          9
     NOPLOT           b
     NOSCALE          1

2nd data card         VULCAN      1         KILL     b
3rd  "     "            "         2          "       b
4th        "            "         3          "       b
5th  "     "            "         4          "       b
6th  "     "            "         10         "       b
7th        "            "         18         "       b
8th  "     "            "         19         "       b
9th  "     "            "         20         "       b
10th "     "            "         21         "       1
```

A partial listing of the printed output for figure 3 follows: (see Figure B)

REMARKS

Note that the phase shifts are defined only to within an integral multiple of pi. It is customary to define $\lim_{k\to\infty} \delta_\ell = 0$. Since the computer delivers values of phase shift only between 0 and 1 in units of pi radians, the question arises as to how to interpret these plots in light of Levinson's theorem. According to Levinson's theorem, the value of the phase shift at zero energy, in units of pi radians, yields the number of ℓ-wave bound states that the well can support, except at certain transitional strengths[3]. If multiples of pi are suppressed in the plot, how can one determine the number of bound states? It will be noted in figure 6 for the deep well, that there are a number of discontinuities in phase. This is a result of the suppression of multiples of n pi. If one plots figure 6 out to energies greater than 400, it will be found that only one more set of discontinuities occurs before the phase shift curves assymptotically approach zero. If one assigns a negative value to jumps up, and a positive value to jumps down, the algebraic number of phase jumps will be equal to the number of ℓ-wave bound states

```
PHASE SHIFTS IN RADIANS FOR A QUANTUM MECHANICAL PARTICLE SCATTERING OFF A 3-D SQUARE WELL.

WELL DEPTH=  10.00000  WELL WIDTH=   1.00000  EZERO=   0.00001 ELAST=  50.00000

   ENERGY(I)   DELTA(0)    DELTA(1)    DELTA(2)    DELTA(3)    DELTA(4)       K          ERGR      OLYMPUS1     HADES1

     0.00001   0.99900     1.00000     0.00000     0.00000     0.00000     0.00316     0.00001  -151.84410  **********
     0.50001   [illegible] 0.86235     0.00056     0.00000     0.00000     0.70711     0.50001   -29.27965    35.80345
     1.00001   0.69864     0.79799     0.00374     0.00004     0.00000     1.00000     1.00001   -15.24113    10.97450
     1.50001   0.63937     0.75005     0.01042     0.00016     0.00000     1.22475     1.50001    -9.87993     4.62578
     2.00001   0.59300     0.71044     0.02172     0.00043     0.00001     1.41422     2.00001    -7.08427     2.13079
     2.50001   0.55521     0.67615     0.03565     0.00091     0.00002     1.58114     2.50001    -5.38946     0.94429
     3.00001   0.52376     0.64566     0.06206     0.00168     0.00004     1.73205     3.00001    -4.26666     0.31989
     3.50001   0.49731     0.61807     0.09236     0.00279     0.00005     1.87083     3.50001    -3.47843    -0.02938
     4.00001   0.47499     0.59282     0.12895     0.00433     0.00015     2.00000     4.00001    -2.90234    -0.22854
     4.50001   0.45617     0.56950     0.16991     0.00636     0.00025     2.12132     4.50001    -2.46699    -0.34211
     5.00001   0.44040     0.54783     0.21205     0.00895     0.00039     2.23607     5.00001    -2.13571    -0.40460
     5.50001   0.42732     0.52760     0.25191     0.01216     0.00058     2.34521     5.50001    -1.87552    -0.43593
     6.00001   0.41662     0.50565     0.29688     0.01606     0.00083     2.44949     6.00001    -1.66996    -0.44772
     6.50001   0.40803     0.49084     0.31573     0.02071     0.00115     2.54951     6.50001    -1.50626    -0.44771
     7.00001   0.40131     0.47407     0.33842     0.02616     0.00155     2.64575     7.00001    -1.37523    -0.44057
     7.50001   0.39622     0.45827     0.35556     0.03245     0.00205     2.73861     7.50001    -1.27015    -0.42943
     8.00001   0.39253     0.44335     0.36809     0.03960     0.00265     2.82843     8.00001    -1.18595    -0.41637
     8.50001   0.39000     0.42927     0.37681     0.04763     0.00337     2.91548     8.50001    -1.11878    -0.40279
     9.00001   0.38842     0.41597     0.38249     0.05653     0.00422     3.00000     9.00001    -1.06565    -0.38964
     9.50001   0.38757     0.40341     0.38576     0.06626     0.00522     3.08221     9.50001    -1.02421    -0.37759
    10.00001   0.38725     0.39157     0.38713     0.07677     0.00637     3.16228    10.00001    -0.99262    -0.36707
    10.50001   0.38728     0.38040     0.38698     0.08796     0.00759     3.24037    10.50001    -0.96938    -0.35839
    11.00001   0.38748     0.36989     0.38563     0.09973     0.00919     3.31663    11.00001    -0.95331    -0.35176
    11.50001   0.38772     0.36001     0.38333     0.11193     0.01058     3.39117    11.50001    -0.94344    -0.34732
    12.00001   0.38787     0.35074     0.38026     0.12440     0.01277     3.46410    12.00001    -0.93901    -0.34517
    12.50001   0.38785     0.34207     0.37660     0.13698     0.01498     3.53554    12.50001    -0.93939    -0.34538
    13.00001   0.38756     0.33397     0.37246     0.14948     0.01721     3.60555    13.00001    -0.94407    -0.34800
    13.50001   0.38702     0.32643     0.36794     0.16175     0.01977     3.67424    13.50001    -0.95267    -0.35309
    14.00001   0.38613     0.31944     0.36313     0.17362     0.02257     3.74166    14.00001    -0.96498    -0.36070
    14.50001   0.38489     0.31298     0.35810     0.18495     0.02561     3.80789    14.50001    -0.98048    -0.37087
    15.00001   0.38331     0.30704     0.35290     0.19565     0.02891     3.87298    15.00001    -0.99931    -0.38348
    15.50001   0.38139     0.30161     0.34758     0.20563     0.03245     3.93701    15.50001    -1.02130    -0.39922
    16.00001   0.37914     0.29667     0.34218     0.21483     0.03624     4.00000    16.00001    -1.04641    -0.41757
    16.50001   0.37658     0.29220     0.33673     0.22323     0.04029     4.06202    16.50001    -1.07469    -0.43890
    17.00001   0.37374     0.28819     0.33127     0.23080     0.04458     4.12311    17.00001    -1.10623    -0.46334
    17.50001   0.37064     0.28463     0.32581     0.23757     0.04910     4.18330    17.50001    -1.14121    -0.49113
    18.00001   0.36730     0.28150     0.32038     0.24354     0.05386     4.24264    18.00001    -1.17988    -0.52251
    18.50001   0.36375     0.27877     0.31499     0.24876     0.05883     4.30116    18.50001    -1.22257    -0.55780
    19.00001   0.36001     0.27642     0.30966     0.25325     0.06400     4.35890    19.00001    -1.26969    -0.59740
    19.50001   0.35612     0.27444     0.30441     0.25707     0.06936     4.41588    19.50001    -1.32180    -0.64180
    20.00001   0.35208     0.27280     0.29924     0.26026     0.07487     4.47214    20.00001    -1.37958    -0.69160
    20.50001   0.34793     0.27148     0.29417     0.26287     0.08053     4.52769    20.50001    -1.44389    -0.74755
    21.00001   0.34369     0.27044     0.28921     0.26493     0.08630     4.58258    21.00001    -1.51583    -0.81060
    21.50001   0.33937     0.26967     0.28436     0.26650     0.09216     4.63681    21.50001    -1.59678    -0.88195
    22.00001   0.33500     0.26914     0.27963     0.26761     0.09808     4.69042    22.00001    -1.68851    -0.96314
    22.50001   0.33058     0.26882     0.27503     0.26831     0.10403     4.74342    22.50001    -1.79331    -1.05514
    23.00001   0.32614     0.26867     0.27056     0.26863     0.10998     4.79583    23.00001    -1.91421    -1.16359
    23.50001   0.32170     0.26864     0.26623     0.26861     0.11590     4.84768    23.50001    -2.05525    -1.28899
    24.00001   0.31725     0.26881     0.26204     0.26828     0.12176     4.89898    24.00001    -2.22196    -1.43714
    24.50001   0.31283     0.26904     0.25800     0.26767     0.12754     4.94975    24.50001    -2.42211    -1.61480
    25.00001   0.30843     0.26934     0.25412     0.26681     0.13320     5.00000    25.00001    -2.66694    -1.93175
    25.50001   0.30407     0.26969     0.25038     0.26573     0.13873     5.04975    25.50001    -2.97336    -2.10272
    26.00001   0.29976     0.27007     0.24680     0.26444     0.14410     5.09902    26.00001    -3.36801    -2.45089
    26.50001   0.29551     0.27044     0.24338     0.26297     0.14929     5.14782    26.50001    -3.89541    -2.91508
    27.00001   0.29132     0.27080     0.24012     0.26134     0.15429     5.19615    27.00001    -4.63605    -3.56543
```

```
PHASE SHIFTS IN RADIANS FOR A QUANTUM MECHANICAL PARTICLE SCATTERING OFF A 3-D SQUARE WELL.

WELL DEPTH=    1.00000  WELL WIDTH=  1.0000000  EZERO=  0.0000001 ELAST= ~~10.0000~~000

    ENERGY(I)    DELTA(0)    DELTA(1)    DELTA(2)    DELTA(3)    DELTA(4) 13.86 K          ERGO       OLYMPUS1    HADES1

    0.0000001    0.0000561   0.0000000   0.0000000   0.0000000   0.0000000   0.0003162   0.0000001   0.3579074  ***********
    0.0140001    0.0208237   0.0000406   0.0000000   0.0000000   0.0000000   0.1183220   0.0140001   0.3577974   5.4614610
    0.0280001    0.0292128   0.0001148   0.0000001   0.0000000   0.0000000   0.1673323   0.0280001   0.3576882   3.8565095
    0.0420001    0.0354937   0.0002106   0.0000002   0.0000000   0.0000000   0.2049393   0.0420001   0.3575800   3.1935058
    0.0560001    0.0406613   0.0003237   0.0000005   0.0000000   0.0000000   0.2366434   0.0560001   0.3574726   2.7831749
    0.0700001    0.0451052   0.0004517   0.0000009   0.0000000   0.0000000   0.2645753   0.0700001   0.3573661   2.5050496
    0.0840001    0.0490272   0.0005929   0.0000013   0.0000000   0.0000000   0.2898277   0.0840001   0.3572605   2.3011525
    0.0980001    0.0525481   0.0007460   0.0000020   0.0000000   0.0000000   0.3130497   0.0980001   0.3571558   2.1437822
    0.1120001    0.0557477   0.0009100   0.0000028   0.0000000   0.0000000   0.3346642   0.1120001   0.3570519   2.0178203
    0.1260001    0.0586519   0.0010842   0.0000037   0.0000000   0.0000000   0.3549649   0.1260001   0.3569490   1.9142263
    0.1400001    0.0613917   0.0012679   0.0000048   0.0000000   0.0000000   0.3741459   0.1400001   0.3568470   1.8272297
    0.1540001    0.0639084   0.0014605   0.0000061   0.0000000   0.0000000   0.3924285   0.1540001   0.3567458   1.7529141
    0.1680001    0.0662566   0.0016616   0.0000076   0.0000000   0.0000000   0.4098782   0.1680001   0.3566454   1.6855814
    0.1820001    0.0684560   0.0018707   0.0000092   0.0000000   0.0000000   0.4266147   0.1820001   0.3565467   1.6322498
    0.1960001    0.0705227   0.0020874   0.0000111   0.0000000   0.0000000   0.4427190   0.1960001   0.3564478   1.5824476
    0.2100001    0.0724701   0.0023114   0.0000131   0.0000000   0.0000000   0.4582577   0.2100001   0.3563503   1.5380593
    0.2240001    0.0743095   0.0025424   0.0000154   0.0000001   0.0000000   0.4732865   0.2240001   0.3562537   1.4982153
    0.2380001    0.0760505   0.0027801   0.0000179   0.0000001   0.0000000   0.4878525   0.2380001   0.3561580   1.4622294
    0.2520001    0.0777015   0.0030242   0.0000206   0.0000001   0.0000000   0.5019961   0.2520001   0.3560633   1.4295517
    0.2660001    0.0792697   0.0032746   0.0000236   0.0000001   0.0000000   0.5157520   0.2660001   0.3559695   1.3997354
    0.2800001    0.0807613   0.0035309   0.0000268   0.0000001   0.0000000   0.5291504   0.2800001   0.3558766   1.3724135
    0.2940001    0.0821820   0.0037930   0.0000302   0.0000001   0.0000000   0.5422178   0.2940001   0.3557846   1.3472813
    0.3080001    0.0835367   0.0040607   0.0000339   0.0000002   0.0000000   0.5549776   0.3080001   0.3556935   1.3240835
    0.3220001    0.0848298   0.0043338   0.0000378   0.0000002   0.0000000   0.5674505   0.3220001   0.3556034   1.3026044
    0.3360001    0.0860654   0.0046122   0.0000420   0.0000002   0.0000000   0.5796552   0.3360001   0.3555143   1.2826621
    0.3500001    0.0872469   0.0048956   0.0000464   0.0000003   0.0000000   0.5916081   0.3500001   0.3554260   1.2640933
    0.3640001    0.0883777   0.0051840   0.0000511   0.0000003   0.0000000   0.6033242   0.3640001   0.3553387   1.2467690
    0.3780001    0.0894607   0.0054772   0.0000561   0.0000003   0.0000000   0.6148171   0.3780001   0.3552524   1.2305661
    0.3920001    0.0904986   0.0057750   0.0000613   0.0000004   0.0000000   0.6260991   0.3920001   0.3551670   1.2153848
    0.4060001    0.0914939   0.0060773   0.0000669   0.0000004   0.0000000   0.6371814   0.4060001   0.3550826   1.2011337
    0.4200001    0.0924488   0.0063840   0.0000727   0.0000005   0.0000000   0.6480741   0.4200001   0.3549991   1.1877332
    0.4340001    0.0933655   0.0066951   0.0000787   0.0000005   0.0000000   0.6587868   0.4340001   0.3549164   1.1751130
    0.4480001    0.0942458   0.0070102   0.0000851   0.0000006   0.0000000   0.6693281   0.4480001   0.3548351   1.1632194
    0.4620001    0.0950915   0.0073295   0.0000916   0.0000007   0.0000000   0.6797059   0.4620001   0.3547545   1.1519696
    0.4760001    0.0959044   0.0076527   0.0000988   0.0000007   0.0000000   0.6899276   0.4760001   0.3546749   1.1413407
    0.4900001    0.0966860   0.0079797   0.0001060   0.0000008   0.0000000   0.7000001   0.4900001   0.3545963   1.1312757
    0.5040001    0.0974376   0.0083106   0.0001136   0.0000009   0.0000000   0.7099296   0.5040001   0.3545186   1.1217433
    0.5180001    0.0981607   0.0086450   0.0001215   0.0000010   0.0000000   0.7197222   0.5180001   0.3544419   1.1126990
    0.5320001    0.0988564   0.0089831   0.0001297   0.0000011   0.0000000   0.7293834   0.5320001   0.3543663   1.1041696
    0.5460001    0.0995261   0.0093246   0.0001381   0.0000012   0.0000000   0.7389162   0.5460001   0.3542916   1.0959440
    0.5600001    0.1001707   0.0096696   0.0001470   0.0000013   0.0000000   0.7483315   0.5600001   0.3542179   1.0881860
    0.5740001    0.1007913   0.0100179   0.0001561   0.0000014   0.0000000   0.7576279   0.5740001   0.3541451   1.0807994
    0.5880001    0.1013889   0.0103694   0.0001655   0.0000015   0.0000000   0.7668116   0.5880001   0.3540734   1.0737625
    0.6020001    0.1019644   0.0107241   0.0001753   0.0000017   0.0000000   0.7758867   0.6020001   0.3540027   1.0670572
    0.6160001    0.1025187   0.0110819   0.0001854   0.0000018   0.0000000   0.7848567   0.6160001   0.3539330   1.0606634
    0.6300001    0.1030526   0.0114427   0.0001958   0.0000019   0.0000000   0.7937255   0.6300001   0.3538643   1.0545532
    0.6440001    0.1035669   0.0118064   0.0002066   0.0000021   0.0000000   0.8024962   0.6440001   0.3537966   1.0487493
    0.6580001    0.1040623   0.0121730   0.0002177   0.0000023   0.0000000   0.8111721   0.6580001   0.3537303   1.0431797
    0.6720001    0.1045395   0.0125425   0.0002291   0.0000024   0.0000000   0.8197561   0.6720001   0.3536643   1.0378673
    0.6860001    0.1049992   0.0129147   0.0002408   0.0000026   0.0000000   0.8282512   0.6860001   0.3535997   1.0327991
    0.7000001    0.1054419   0.0132896   0.0002529   0.0000028   0.0000000   0.8366601   0.7000001   0.3535361   1.0279362
    0.7140001    0.1058684   0.0136671   0.0002654   0.0000030   0.0000000   0.8449853   0.7140001   0.3534736   1.0232944
    0.7280001    0.1062791   0.0140472   0.0002782   0.0000032   0.0000000   0.8532292   0.7280001   0.3534120   1.0188542
    0.7420001    0.1066747   0.0144298   0.0002913   0.0000034   0.0000000   0.8613943   0.7420001   0.3533516   1.0146060
    0.7560001    0.1070556   0.0148149   0.0003048   0.0000036   0.0000000   0.8694827   0.7560001   0.353[illegible]  [illegible]
    [illegible]  0.1074216   0.0735264   0.00[illegible]  0.000[illegible]  0.0[illegible]
```

in the well, for each curve.

9.0 ACKNOWLEDGEMENT

The author is indebted to Professor P. H. E. Meijer, who suggested this work, and to Mr. S. Brenner, who contributed subroutines LAYOUT, SCALE and AXIS, to Mrs. J. Mason, who contributed subroutines SPHJN and SPHYN, and to Dr. A. H. Lowrey, for helpful discussions.

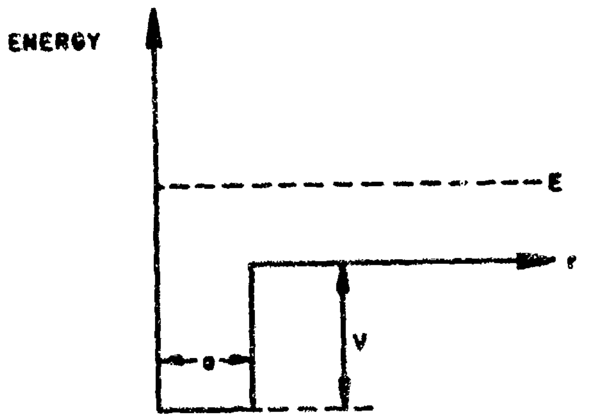

Fig. 1 — The square well potential

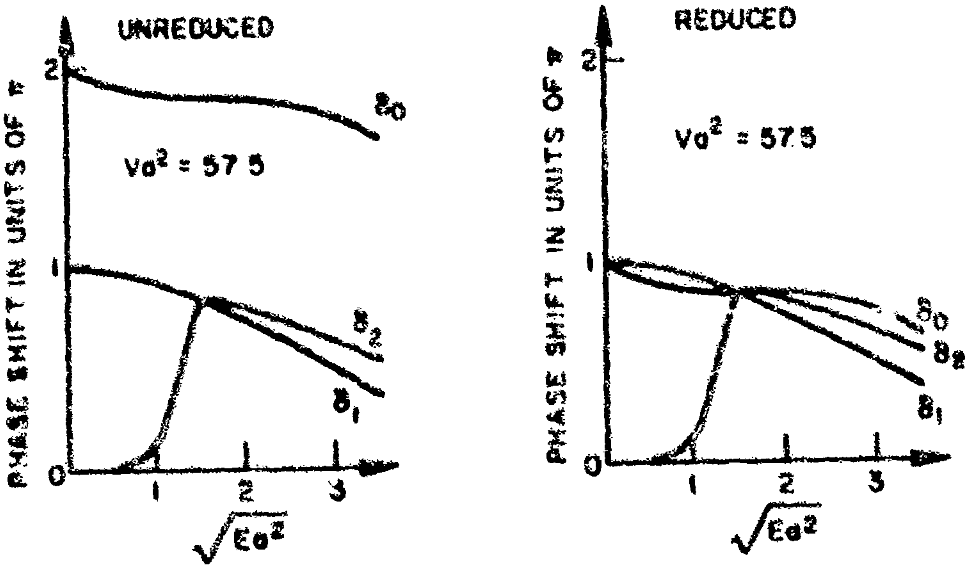

Fig. 2 — Deep well phase shift, unreduced and reduced

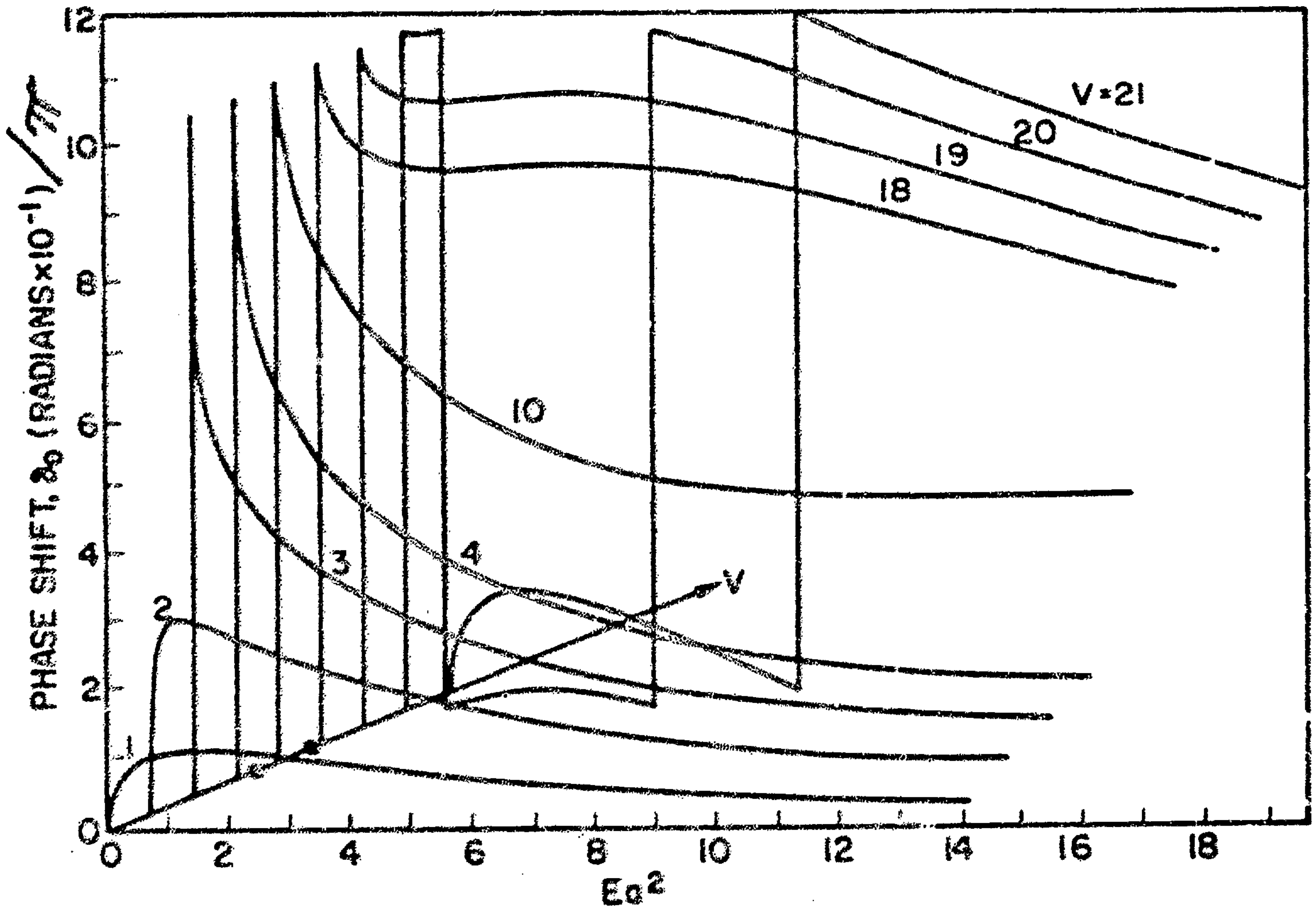

Fig. 3 — Reduced phase shift for $\ell = 0$ for various values of the well depth V, in units δ_0/π, where δ_0 is expressed in radians, and where a is taken as unity

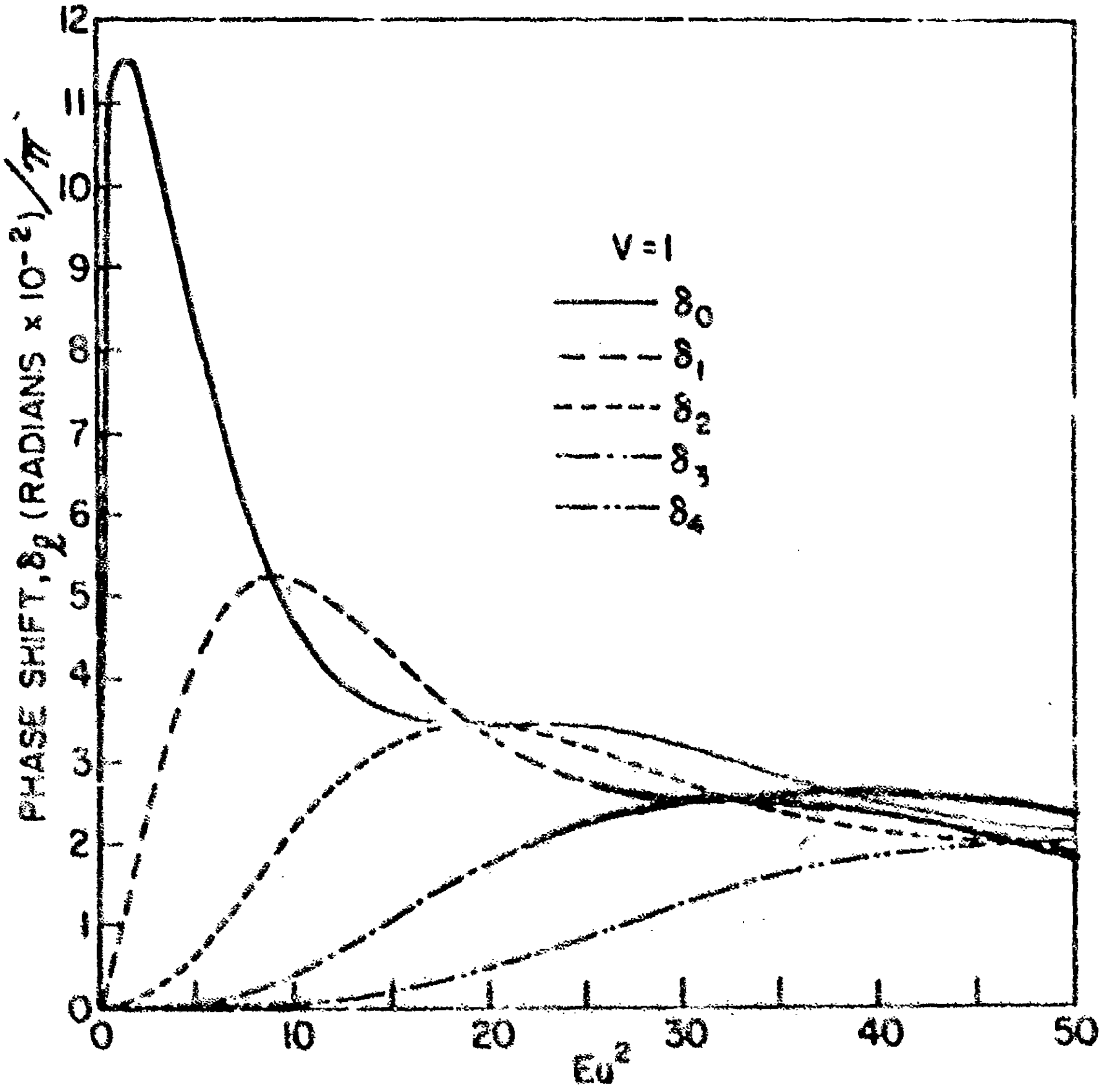

Fig. 6 — Shallow well phase shifts for ℓ = 1,2,3,4,5 in units δ_ℓ/π, where δ_ℓ is expressed in radians, and where a is taken as unity

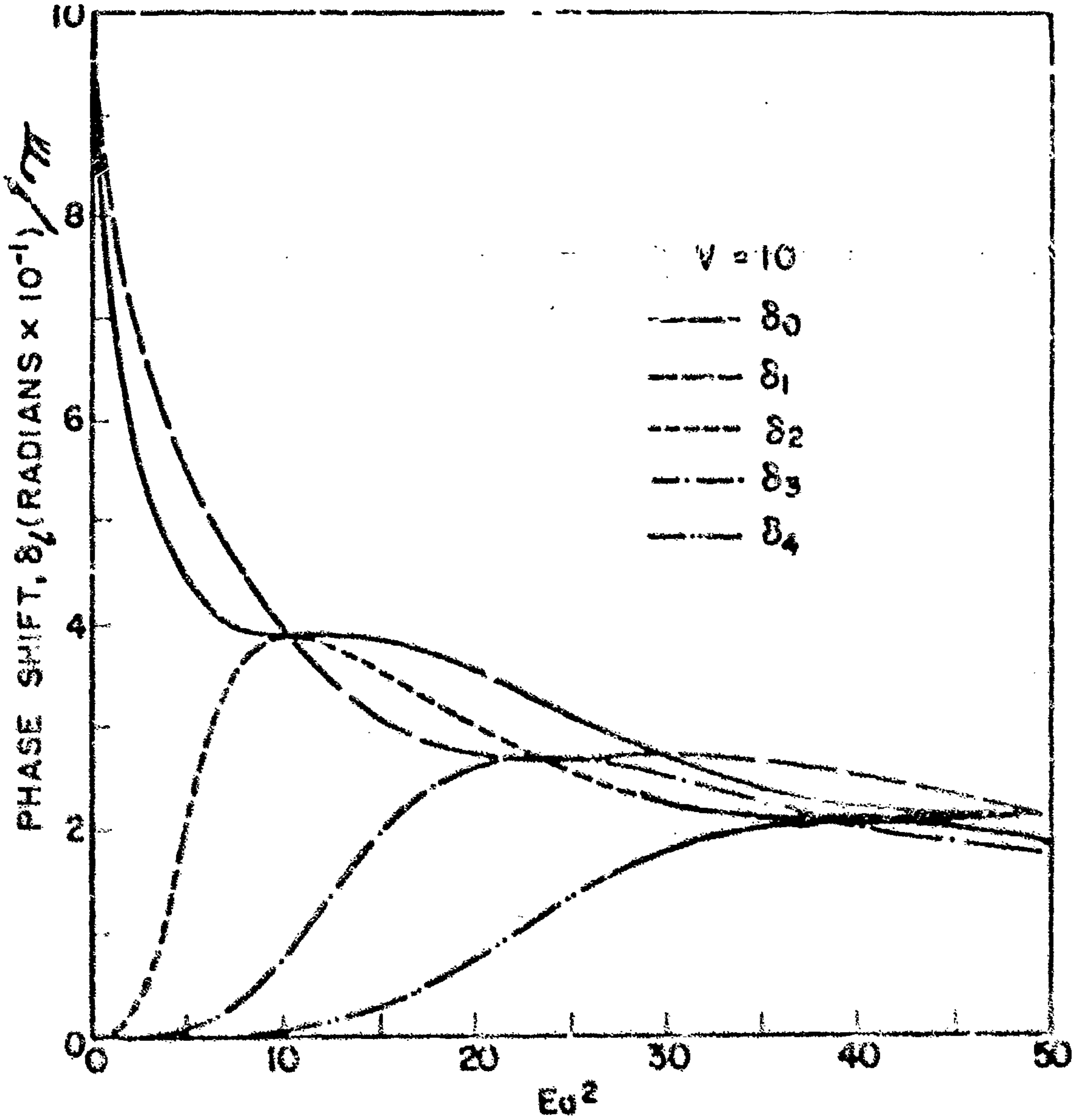

Fig. 6 – Medium well phase shifts for ℓ = 0,1,2,3,4 in units δ_ℓ/π, where δ_ℓ is expressed in radians, and where a is taken as unity

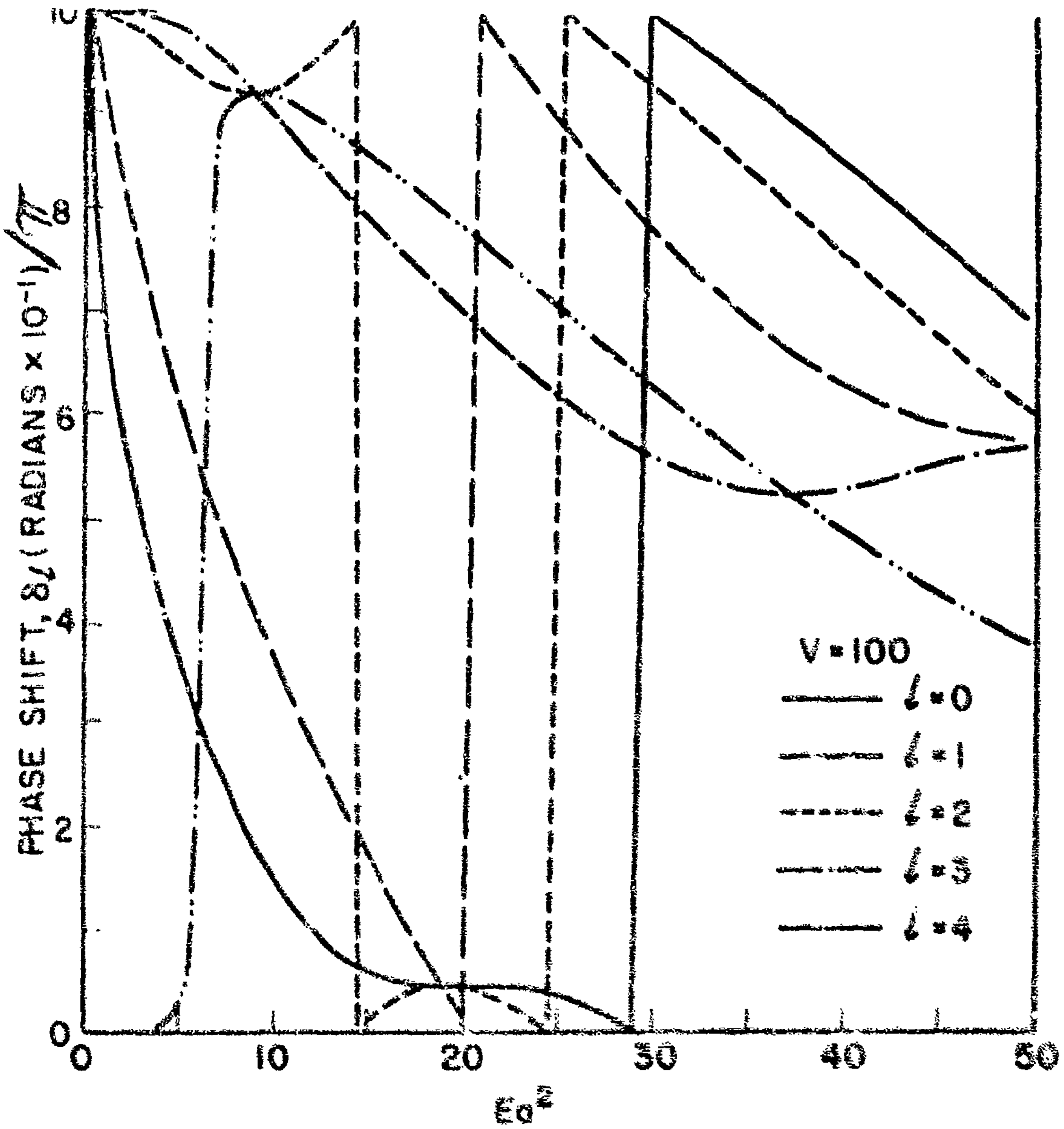

Fig. 6 — Deep well reduced phase shifts for ℓ = 0,1,2,3,4 in units δ_ℓ/π, where δ_ℓ is expressed in radians, and where a is taken as unity

2-D Version

```
      PROGRAM PHASHIFT
C  TWO   DIMENSIONAL VERSION
C THIS PROGRAM CALCULATES PHASE SHIFTS IN RADIANS OR DEGREES VS E*A**2
C OR K OR E IN DIMENSIONLESS UNITS FOR THE QUANTUM MECHANICAL SCATTERING
C OF A PARTICLE OF ENERGY E FROM A 3-D SQUARE WELL OF WIDTH A AND DEPTH
C -V,  THE INPUT PARAMETERS ARE ( 1). EZERO (THE LOWEST ENERGY VALUE) CO
COLS 1-10.(2). EPSILON(THE ENERGY INCREMENT) COLS11-20. (3). /ULCAN
C  THE ABSOLUTE VALUE OF THE WELL DEPTH) COLS 21-30
C  (4). FERMI (THE WELL WIDTH) COLS 31-40.  (5).  MAX (THE
C  HIGHEST ORDER PHASE SHIFT DESIRED.  FOR S,P,D,F) MAX=1,2,3,4. COL 45
C  (6).  KILL(NONZERO ONLY ON LAST DATA CARD) COL 72.  (7). LAST(THE TOT
C  AL NUMBER OF POINTS DESIRED TO BE CALCULATED). MAX. 1000. COLS 55-58
C  (8). JELLO (IF PRINTOUT OF PHASE SHIFTS IN DEGREES IS DESIRED, ENTER
C    1 IN COL 50, IF RADIANS DESIRED, LEAVE COL 50 BLANK) COL50
C  IF NO PLOT IS DESIRED, ENTER A 1 IN COL. 65.  IF ORDINAL SELF-SCALING
C DESIRED, ENTER A 1 IN COL 70, AND INSERT VALUES OF YMIN AND YMAX  BEFO
C  T CALL.
      COMMON/LABEL/IL,ILX(5),ILY(5),ILH(5),ILR(5),ILC(5),ILABLE(5,5)
      COMMON /RANGE/ XMAX,XMIN,YMAX,YMIN,IFRST(5),ILST(5),XSIZE
      REAL ILX,ILY,ILH,ILR
      REAL N
      REAL NZ
      COMMON/1/ DELTA(1000),CASTOR(1000),POLLUX(1000),HERCULES(1000),AJA
     1X(1000),OLYMPUS(1000),HADES(1000),B(1000),DB(1000),N(1000),DN(1000
     2),BZ(1000),DBZ(1000),NZ(1000),GAMMA(1000),ENERGY(1000),CYCLOPS(100
     30),CENTAUR(1000),ZFUS(1000),RHO(1000),DNZ(1000)
      DIMENSION              DELTA1(1000),DELTA2(1000),DELTA3(1000),DELTA
     14(1000),W(5)
      DIMENSION ERGO(1000)
      COMMON/A/ PHASE(5000),ERGON(5000)
    9 READ (60,10) EZERO,EPSILON,VULCAN,FERMI,MAX,JELLO,LAST,NOPLOT,NOSC
     1ALF,KILL
   10 FORMAT(4(F10.5),4X,I1,4X,I1,4X,I4,6X,I1,4X,I1,1X,I1)
      IFRST(1)=1
      IFRST(2)=1+   LAST
      IFRST(3)=1+ 2*LAST
      IFRST(4)=1+ 3*LAST
      IFRST(5)=1+ 4*LAST
      ILST(1)=  LAST
      ILST(2)=2*LAST
      ILST(3)=3*LAST
      ILST(4)=4*LAST
      ILST(5)=5*LAST
      IF (JELLO.EQ.1) GO TO 2
    1 PRINT 6
      GO TO 3
    2 PRINT 7
    3 PRINT 12,VULCAN,FERMI,EZERO,XMAX
      PRINT 8
    6 FORMAT (1X,*PHASE SHIFTS IN RADIANS FOR A QUANTUM MECHANICAL PARTI
     1CLE SCATTERING OFF A 3-D SQUARE WELL.*/)
    7 FORMAT (1X,*PHASE SHIFTS IN DEGREES FOR A QUANTUM MECHANICAL PARTI
     1CLE SCATTERING OFF A 3-D SQUARE WELL.*/)
    8 FORMAT(5X,*ENERGY(I)*4X*DELTA(0)*4X*DELTA(1)*4X*DELTA(2)*4X*DELTA(
     13)*4X*DELTA(4)*4X*    K    *4X*  ERGO  *4X*OLYMPUS1*4X*HADES1*/)
   12 FORMAT(1X,*WELL DEPTH= *F10.5,*  WELL WIDTH= *F10.5,*  EZERO= *
     1 F10.5,* FLAST= *F10.5/)
      K=MAX+1
      ENERGY(1)=EZERO
      DO 91 I=1,LAST
      ERGO(I)=ENERGY(I)*(FERMI)**2
   14 ENERGY(I+1)=EPSILON+ENERGY(I)
   18 CYCLOPS(I)=SQRT(ENERGY(I))
   20 CENTAUR(I)=SQRT(ENERGY(I)+VULCAN)
   25 ZEUS(I)=CYCLOPS(I)*FERMI
      ZEUSS=ZEUS(I)
   27 IF(ZEUSS.EQ.0) 78,30
   30 RHO(I)=CENTAUR(I)*FERMI
      RHOO=RHO(I)
   34 CALL SPHJN(RHOO,K,B)
      CALL SPHYN(RHOO,K,N)
      DB(1)=-B(2)
      DN(1)=-N(2)
      DO 35 JI=2,MAX
      DB(JI)=(JI*B(JI-1)-(JI+1)*B(JI+1))/(2*JI+1)
      DN(JI)=(JI*N(JI-1)-(JI+1)*N(JI+1))/(2*JI+1)
   35 CONTINUE
      CALL SPHJN(ZEUSS,K,BZ)
      CALL SPHYN(ZEUSS,K,NZ)
      DBZ(1)=-BZ(2)
      DNZ(1)=-NZ(2)
   37 DO 38 KI=2,MAX
      DBZ(KI)=(KI*BZ(KI-1)-(KI+1)*BZ(KI+1))/(2*KI+1)
      DNZ(KI)=(KI*NZ(KI-1)-(KI+1)*NZ(KI+1))/(2*KI+1)
   38 CONTINUE
   40 DO 76 LI=1,MAX
   44 GAMMA(LI)=CENTAUR(I) *DB(LI)/B(LI)
   45 CASTOR(LI)=CYCLOPS(I)*DBZ(LI)
   50 POLLUX(LI)=GAMMA(LI)*BZ(LI)
   55 HERCULES(LI)=CYCLOPS(I)*DNZ(LI)
   60 AJAX(LI)=GAMMA(LI)*NZ(LI)
   65 OLYMPUS(LI)=CASTOR(LI)-POLLUX(LI)
   70 HADES(LI)=HERCULES(LI)-AJAX(LI)
   75 W(LI)   =ATAN2(OLYMPUS(LI),HADES(LI))
      IF(W(LI).LT.0) W(LI)=W(LI)+3.14159
       IF(JELLO.EQ.0) GO TO 755
      W(LI)=W(LI)*57.29578
      IF(JELLO.EQ.1)GO TO 76
  755 W(LI)=W(LI)/3.14159
   76 CONTINUE
   77 GO TO 80
   78 DO 79 LI=1,MAX
      W(LI)=0.
   79 CONTINUE
      OLYMPUS(1)=0.
      HADES(1)=0.
   80 PRINT 85,ENERGY(I),W(1),W(2),W(3),W(4),W(5),ZEUS(I),ERGO(I),OL
     1YMPUS(1),HADES(1)
   85 FORMAT (10(2X,F10.5))
      DELTA(I)=W(1)
      DELTA1(I)=W(2)
      DELTA2(I)=W(3)
      DELTA3(I)=W(4)
      DELTA4(I)=W(5)
   91 CONTINUE
   86 PRINT 87
   87 FORMAT (///)
      IF(NOPLOT.EQ.1) GO TO 95
      IF (NOSCALE.EQ.1)GO TO 99
      CALL MINIMAX(DELTA,YMIN,YMAX,LAST)
   99 XSIZE=10.
      XMIN=ERGO(1)
      XMAX=ERGO(LAST)
C IF SELF-SCALING NOT USED8 THEN YMIN AND YMAX VALUES MUST BE SUPPLIED.
      YMIN=0.
      YMAX=1.
C IF DESIRED, DIFFERENT ORDER PHASE SHIFTS MAY BE PLOTTED ON THE SAME
C   GRAPH.  USE NOSCALE TO BYPASS, PUT A 1 IN COL 70.  SUPPLY YMIN,YMAX.
      NLAST=5*LAST
      J=LAST
      DO 90 M=1,J
      PHASE(M)=DELTA(M)
      PHASE(M+J)=DELTA1(M)
      PHASE(M+2*J)=DELTA2(M)
      PHASE(M+3*J)=DELTA3(M)
      PHASE(M+4*J)=DELTA4(M)
      ERGON(M)=ERGO(M)
      ERGON(M+J)=ERGO(M)
      ERGON(M+2*J)=ERGO(M)
      ERGON(M+3*J)=ERGO(M)
      ERGON(M+4*J)=ERGO(M)
   90 CONTINUE
      CALL LAYOUT(ERGON,PHASE,NLAST,6HE*A**2,7H-DELTA-,6,7,5)
      PRINT 100
  100 FORMAT(///)
   95 IF(KILL.EQ.0) GO TO 9
      CALL STOPPLOT
  190 END
      SUBROUTINE MINIMAX(ARRAY,YMIN,YMAX,LAST)
      DIMENSION ARRAY(1000)
C   THIS SUBROUTINE FINDS BOTH THE LARGEST AND SMALLEST NUMBERS IN ARRAY.
C LARGEST NUMBER
      J=1
      JJ=1+J
   10 DO 30 K=JJ,LAST
      IF(ARRAY(J).GE.ARRAY(K))30,20
   20 J=K
      IF(K.EQ.LAST) GO TO 40
       GO TO 50
   30 CONTINUE
   40 YMAX=ARRAY(J)
      PRINT 45,JJ
   45 FORMAT (1X,*JJ=*I3)
      GO TO 60
   50 JJ=J+1
      GO TO 10
C SMALLEST NUMBER
   60 L=1
      LL=1+L
  100 DO 300 M=LL,LAST
      IF(ARRAY(L).LE.ARRAY(M))300,200
  200 L=M
      IF(M.EQ.LAST)GO TO 400
      GO TO 500
  300 CONTINUE
  400 YMIN=ARRAY(L)
      PRINT 450,LL
  450 FORMAT (1X,*LL=*I3)
      GO TO 600
  500 LL=L+1
      GO TO 100
  600 RETURN
      END
      SUBROUTINE SPHJN(X,N,ARR)
C3 NRL BESN       BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS
      DIMENSION ARR(1)
      RX=1.0E+307
      NF=1
      NN=N+1
      IF(X.EQ.0.0) GO TO 4
      IF(X.LE.0.5)GO TO 6
C     FOR ORDERS LESS THAN ARGUMENT, AND ARGUMENT GREATER THAN 0.5
      ARR(1)=SIN(X)/X
      ARR(2)=ARR(1)/X-COS(X)/X
      NM=X-2
      IF(N.LT.X)NM=N-1
      IF(NM.LT.0)NM=0
      DO 16 I=1,NM
      FN=I+1
      ARR(I+2)=(FN+1.0)*ARR(I+1)/X-ARR(I)
C     CHECK FOR UNDERFLOW
      IF(ARR(I+2).NE.0.0)GO TO 16
      ARR(I+1)=ARR(I+1)*RX
      ARR(I+2)=(FN+1.0)*ARR(I+1)/X-ARR(I)*RX
      PRINT 13, X,I,N
   16 CONTINUE
      IF(N.LT.X)RETURN
      HOLD=ARR(NM+2)
C     FOR ORDERS GREATER THAN OR LQUAL TO ARGUMENT, AND ARGUMENT
C     GREATER THAN 0.5
   15 ALPHA=1.0
      M=-2-NM
      NOTE=0
      I=N+47
      IF(X/N.GT.0.9)I=I+X/10.0
      ARR(I+2)=ALPHA
      ARR(I+3)=0.0
      I=-I-1
      DO 2 KK=I,M
      K=-KK
      NOO=K+1
      TKP1=K+K+1
      ARR(K)=TKP1*ARR(K+1)/X-ARR(K+2)
      IF EXPONENT FAULT 18,2
   18 TEMPY=ARR(K)=ARR(K+1)/RX*TKP1/X-ARR(K+2)/RX
      ARR(K)=ARR(K)*RX
      IF EXPONENT FAULT 24,2
   24 ARR(K)=TEMPY
      ARR(K+1)=ARR(K+1)/RX
      PRINT 13,X,N,NOO
    2 CONTINUE
   12 C=HOLD/ARR(K)
      IF(C.NE.0.0)GO TO 14
      HOLD=HOLD*RX
      NOTE=NOTE+1
      GO TO 12
   14 IF(NOTE.EQ.0)GO TO 17
      MULT=307*NOTE
      NUL=L-1
      PRINT 21,X,N,NUL,MULT
   21 FORMAT (3X*FOR ARGUMENT OF *F17.10* AND ORDERS OF *I4* TO *I4*
    1        *SPH.IN EXPONENT WAS INCREASED BY *I6)
   17 DO 3 I=K,NN
      ARR(I)=ARR(I)*C
    3 CONTINUE
      RETURN
    4 ARR(1)=1.0
      DO 5 L=2,NN
      ARR(L)=0.0
    5 CONTINUE
      RETURN
C     SERIES FOR ARGUMENT LESS THAN OR EQUAL TO 0.5
    6 EPS=5.0E-11
      CH=CS=DN=DIV=1.0
      DO 10 L=1,NN
      ITEXP=NEXPA=MEXPA=LEXPA=0
      NO=L-1
      IF(L.EQ.1) GO TO 7
      CS=CS*X/DN
C     CHECK FOR UNDERFLOW
      IF(CS.NE.0.0) GO TO 7
C     SCALE FOR THIS N AND ALL FOLLOWING N
      CS=CH*RX*X/DN
      NEXPA=NEXPA+1
    7 DN=DN+2.0
      CH=CS
      XKK=0.0
      A=CH2=FORTT=1.0/DIV
      TN=2.0*NO
      FN=-0.5*X*X
    8 XKK=XKK+1.0
      FD=XKK*(TN+XKK+XKK+1.0)
      A=A*FN/FD
C     CHECK FOR UNDERFLOW
      IF(A.NE.0.0)GO TO 25
C     SCALE FOR THIS N AND ALL FOLLOWING N
      A=CH2*RX*FN/FD
      DIV=DIV/RX
      MEXPA=MEXPA+1
      FORTT=FORTT*RX
   25 CH2=A
      TT=A+FORTT
      IF(TT.EQ.0.0.OR.ABS(A/TT).LT.EPS) GO TO 9
      FORTT=TT
      GO TO 8
    9 ARR(L)=CS*TT
C     CHECK FOR UNDERFLOW
      IF(ARR(L).NE.0.0)GO TO 31
C     SCALE FOR THIS N AND ALL FOLLOWING N
      ARR(L)=CS*RX*TT
      DIV=DIV/RX
      LEXPA=LEXPA+1
   31 ITEXP=NEXPA+MEXPA+LEXPA
      IF(ITEXP.EQ.0)GO TO 10
      IF(ITEXP.EQ.1)GO TO 29
      PRINT 30, MEXPA,MEXPA,LEXPA
   30 FORMAT(3X,* SCALE ERROR *3(2X,I10))
   20 PRINT 13, X,NO,N
   13 FORMAT(3X,*FOR ARGUMENT OF *,F17.10,* AND ORDERS OF *,I4,* TO *,
     1 I4,* SPHJN EXPONENT WAS INCREASED BY 307*)
   10 CONTINUE
      END
      SUBROUTINE SPHYN(X,N,ARR)
      DIMENSION ARR(1)
      RX=1.0E+307
      IF(X.EQ.0.)5,3
    5 PRINT 4
    4 FORMAT(3X,*DO NOT USE ZERO ARGUMENT FOR SPHYN*)
      RETURN
C     FIND Y0 AND Y1
    3 ARR(1)=-COS(X)/X
      ARR(2)=ARR(1)/X-SIN(X)/X
      NM1=N-1
      DO 2 K=1,NM1
      TKP1=K+K+1
      ARR(K+2)=TKP1*ARR(K+1)/X-ARR(K)
C     CHECK FOR OVERFLOW
      IF EXPONENT FAULT 8,2
    8 ARR(K+1)=ARR(K+1)/RX
      ARR(K+2)=TKP1*ARR(K+1)/X-ARR(K)/RX
      PRINT 7,X,K,N
    7 FORMAT(3X,*FOR ARGUMENT OF *,F17.10,* AND ORDERS OF *,I4,* TO *,
     1 I4,* SPHYN EXPONENT WAS DECREASED BY 307*)
    2 CONTINUE
      END
      SUBROUTINE LAYOUT (XBEFR,YBEFR,NN,XBCD,YBCD,NNX,NNY,NUMBER)
      COMMON/LABEL/IL,ILX(5),ILY(5),ILH(5),ILR(5),ILC(5),ILABLE(5,5)
      COMMON /RANGE/ XMAX,XMIN,YMAX,YMIN,IFRST(5),ILST(5),XSIZE
      REAL ILX,ILY,ILH,ILR
      DIMENSION XBCD(5),YBCD(5)
      DIMENSION XBEFR(NN),YBEFR(NN),PLTARRAY(254)
      DATA (ISTRT=1)
C               INITIALIZATION
      N=NN
      NX=NNX
      NY=NNY
      IF (XSIZE.EQ.0.0) XSIZE=10.
      YSIZE=10.
      NSVE=N
      N=IABSF (N)
      IF (NUMBER.EQ.0) NUMBER=1
      IF (ISTRT.EQ.2) GO TO 2
      CALL PLOTS(PLTARRAY,254,13)
      ISTRT=2
    2 CONTINUE
C
C               LABEL X AXIS
C
C

      DX= (XMAX-XMIN)/XSIZE
      IF (DX.NE.0) GO TO 3
      CALL SCALE (XBEFR,N,XSIZE,XMIN,DX,1)
      GO TO 7
    3 CONTINUE
```

```
      DO 6 I=1,N
C     PRINT 5555,XBFFR(I),YBFFR(I),I
 5555 FORMAT (2F20.10,I10)
    6 XBFFR(I)= (XBFFR(I)-XMIN)/DX
    7 CONTINUF


C     CALL SCALF (XBFFR,N,XSIZE,XMIN,DX,1)
      ZFRO=XSIZE-ZERO
      IF (NSVF) 5,10,10
    5 XMIN=XSIZF*DX-ABSF(XMIN)
      DX=-DX
   10 CONTINUF
      ZFRO=(0.0-XMIN)/DX
      PRINT 4,DX,XMIN,ZERO
    4 FORMAT (* DX=*,3F20.5)
      CALL AXIS (0.0,0.0,XBCD,NX,XSIZF,0.0,XMIN,DX)
      PRINT 4,DX,XMIN,ZFRO
      NX=-NX
      CALL AXIS (0.0,YSIZF,XBCD,NX,XSIZF,0.0,XMIN,DX)
C
      PRINT 4,DX,XMIN,ZFRO
C                LABFL Y AXIS
C


      DY= (YMAX-YMIN)/YSIZE
      IF (DY.NE.0.0) GO TO 14
      CALL SCALF (YBFFR,N,YSIZF,YMIN,DY,1)
      GO TO 15
   14 CONTINUF
      DO 16 I=1,N
   16 YBFFR(I)=(YBFFR(I)-YMIN)/DY
   15 CONTINUF

      PRINT 11,DY,YMIN
   11 FORMAT (* DY=*,3F20.4)
      CALL AXIS (0.0,0.0,YBCD,NY,YSIZF,90.,YMIN,DY)
      NY=-NY
      PRINT 11,DY,YMIN
      CALL AXIS (XSIZF,0.0,YBCD,NY,YSIZF,90.,YMIN,DY)
      PRINT 11,DY,YMIN
C
C               PLOT CURVF
      NUMBER1=NUMBFR
      IF (NUMBER.LT.0) NUMBFR=-NUMBER
      ND=(N/NUMBFR)-1
      N2=0
      IF (NUMBER.NE.1) GO TO 8
      IFRST(1)=1
      ILST(1)=N
    8 CONTINUE
      DO 150 J=1,NUMBER
      IF (NUMBFR1.LT.0) GO T) 148
      N1=IFRST(J)
      N2=ILST(J)
      GO TO 149
  148 CONTINUF
      N1=N2+1
      N2=N1+ND
  140 CONTINUE
      IPN=3
C
C     CALL PLOT (XBFFR(1),YBFFR(1),3)
C
      DO 100 I=N1,N2
C     PRINT 5555,XBFFR(I),YBFFR(I),I
      XBF= XBFFR(I)
      YBF= YBFFR(I)
      IF (NSVF.LT.0) XBFFR(I)=XSIZE-XBFFR(I)
      IF (XBF.LT.0.0.OR.XBF.GT.XSIZE) GO TO 100
      IF (YBF.LT.0.0.OR.YBF.GT.YSIZE) GO TO 100
      CALL PLOT(XBF,YBF,IPN)
      IPN=2
  100 CONTINUE
  150 CONTINUE
      IF (IL.EQ.0) GO TO 255
      ICT=0
  252 ICT=ICT+1
      IF (ICT.GT.IL) GO TO 255
      CALL SYMBOL(ILX(ICT),ILY(ICT),ILH(ICT),ILABLE(1,ICT),ILR(ICT)
     1,ILC(ICT))
      GO TO 252
  255 CONTINUE
      CALL PLOT (XSIZE+3.,0.0,-3)
      PRINT 250
  250 FORMAT (* PLOT DONE*)
C
C

C         RESTORE THE DATA
      DX=ABS(DX)
      DY=ABS(DY)
C
      DO 400 I=1,N
      XBFFR(I)=(DX*XBFFR(I))+XMIN
      YBFFR(I)=(DY*YBFFR(I))+YMIN
  400 CONTINUE
C
      XMAX=0.0
      XMIN=0.0
      YMAX=0.0
      YMIN=0.0
C
      RETURN
      END
      SUBROUTINE SCALE (X,N,S,YMIN,DY,K)
      DIMENSION X(2)
      YMAX=X(1)
      YMIN=YMAX
      NP=N*K
      DO 10 I=1,NP,K
      IF(YMAX-X(I))5,6,6
    5 YMAX=X(I)
    6 IF(X(I)-YMIN) 7,10,10
    7 YMIN=X(I)
   10 CONTINUE
      DY=(YMAX-YMIN)/ (S)
      DO 20 I=1,N
   20 X(I)=(X(I)-YMIN)/DY
      PRINT 25,YMIN,YMAX,DY,X(1),X(N)
   25 FORMAT(5F25.5)
      RETURN
      END
      SUBROUTINE AXIS (X,Y,BCD,NC,SIZE,THETA,YMIN,DY)
      X14=.07
      SIGN=1.0
      IF (NC) 1,2,2
    1 SIGN=-1.0
    2 NAC=XABSF(NC)
      TH=THETA*0.017453294
      N=SIZE+.50
      CTH=COSF(TH)
      STH=SIN(TH)
      TN=N
      XB=X
      YB=Y
      XA=X-.1*SIGN*STH
      YA=Y+.1*SIGN*CTH
C     IF (THETA-90.)10,12,12
C  12 XA=-XA
   10 CALL PLOT (XA,YA,3)
C
C DO LOOP DRAWS AXIS AND MAKES TIC MARKS
C
      DO 20 I=1,N
      CALL PLOT (XB,YB,2)
      XC=XB+CTH
      YC=YB+STH
      CALL PLOT (XC,YC,2)
      XA=XA+CTH
      YA=YA+STH
      CALL PLOT (XA,YA,2)
      XB=XC
   20 YB=YC
      CHAR=ABSF (YMIN)
      ABSV= ABSF (YMIN+DY)
      IF (ABSV-CHAR) 5,6,6
    5 ABSV=CHAR
    6 FXP=0.0
   89 NT=ABSV+0.1
      CHAR=NT
      IF (CHAR-ABSV) 90,92,92
   90 FXP=FXP-1.0
      ABSV=ABSV*10.
      GO TO 89
   91 FXP=FXP+1.0
    2 ABSV=ABSV/10.0
      NT=ABSV
      CHAR=NT
C     IF (CHAR-ABSV)93,91,91
      IF (ABSV.GT.1) GO TO 91
   93 ADY=DY*10.0**(-FXP)
      ABSV=YMIN*10.0**(-FXP)+TN*ADY
      XA=XB-(.20*SIGN-0.05)*STH-.25*CTH
      YA=YB+(.20*SIGN-.05)*CTH-.25*STH
      N=N+1
```

```
C      IF (THETA-90.) 31,32,32
C   32 XA=-XA+.1
C
C DO LOOP PLOTS NUMBERS ALONG AXIS
C
   31 DO 30 I=1,N
      CALL NUMBER (XA,YA,.07,ABSV,THETA,4HF7.3)
      ABSV=ABSV-ADY
C      IF (ABSV.LT.0.AND.ABSV.GT.-1) 55,56
C   55 ABSV=0
   56 XA=XA-CTH
   30 YA=YA-STH
      TNC=NAC+7
C
C CALCULATES CENTER POSITION FOR LABEL
      XA=X+(SIZE/2.0-.06*TNC)*CTH-(-.07+SIGN*.36)*STH
      YA=Y+(SIZE/2.0-.06*TNC)*STH+(-.07+SIGN*.36)*CTH
C      IF (THETA-90.) 41,42,42
C   42 XA=-XA+.1
   41 CALL SYMBOL (XA,YA,X14,BCD,THETA,NAC)
C
C PRINTS LABEL
      XA=XA+((TNC-6.0)*0.12)*CTH
      YA=YA+((TNC-6.0)*0.12)*STH
      IF (EXP) 35,50,35
   35 CALL SYMBOL (XA,YA,X14,8H(X10    ),THETA,8)
      XA=XA+.48*CTH-.07*STH
      YA=YA+.48*STH+.04*CTH
      IF (EXP) 40,50,40
C
C    PRINTS SCALE FACTOR
   40 CALL NUMBER (XA,YA,X14,EXP,THETA,4HF3.0)
   50 CONTINUE
      CALL PLOT (XA,YA,3)
      RETURN
       END
//GG
//GG
//GG
```

3-D Version

```
      PROGRAM PHASHIFT
C  THREF DIMENSIONAL VERSION
C THIS PROGRAM CALCULATES PHASE SHIFTS IN RADIANS OR DEGREES VS E*A**2
C OR K OR E IN DIMENSIONLFSS UNITS FOR THE QUANTUM MECHANICAL SCATTERING
C OF A PARTICLE OF ENERGY E FROM A 3-D SQUARE WELL OF WIDTH A AND DEPTH
C -V.
C THF 1ST DATA CARD CONTAINS THF FOLLOWING PARAMETERS...
C      THE INPUT PARAMETERS ARE ( 1). EZERO (THE LOWEST ENERGY VALUE) CO
COLS 1-10.(2). EPSILON(THE ENERGY INCREMENT) COLS11-20. (3). SKIP TO 4
C  (4). FERMI (THE WELL WIDTH) COLS 21-30.  (5).  MAX (THE
C  HIGHEST ORDER PHASF SHIFT DESIRFD.  FOR S,P,D,F) MAX=1,2,3,4. COL 45
C  (6).            SKIP TO 7                         (7). LAST(THE TOT
C  AL NUMBER OF POINTS DFSIRED TO BF CALCULATED). MAX. 1000. COLS 55-58
C  (8). JELLO (IF PRINTOUT OF PHASE SHIFTS IN DFGREES IS DESIRED, ENTER
C    1 IN COL 50, IF RADIANS DESIRFD, LFAVE COL 50 BLANK) COL50
C ENTFR THE NUMBER OF CURVES OT BF PLOTTED IN COL 60(MAX 9).
C  IF NO PLOT IS DESIRED, ENTFR A 1 IN COL. 65.  IF ORDINAL SELF-SCALING
C DESIRED, ENTFR A 1 IN COL 70, AND INSFRT VALUFS OF YMIN AND YMAX  BEFO
C  T CALL.
C THF 2ND DATA CARD CCNTAINS THF WELL DEPTH VULCAN, COLS 1-10, AND KILL
C  COL 21,DESIGNATFS THE LAST DATA CARD.
      COMMON/LABEL/IL,ILX(9),ILY(9),ILH(9),ILR(9),ILC(9),ILABLE(9,9)
      COMMON /RANGE/ XMAX,XMIN,YMAX,YMIN,IFRST(9),ILST(9),XSIZE
      REAL ILX,ILY,ILH,ILR
      REAL N
      REAL NZ
      COMMON/1/ DELTA(1000),CASTOR(1000),POLLUX(1000),HERCULES(1000),AJA
     1X(1000),OLYMPUS(1000),HADES(1000),B(1000),DB(1000),N(1000),DN(1000
     2),BZ(1000),DBZ(1000),NZ(1000),GAMMA(1000),ENERGY(1000),CYCLOPS(100
     30),CENTAUR(1000),ZFUS(1000),RHO(1000),DNZ(1000)
      DIMENSION            DELTA1(1000),DELTA2(1000),DELTA3(1000),DELTA
     14(1000),W(5)
      DIMENSION ERGO(1000)
      COMMON/3/ PHASE(9000),ERGON(9000)
      KC=-1
  700 READ (60,800)EZERO,EPSILON,FERMI,MAX,JELLO,LAST,NCURVES,NOPLOT,NOS
     1CALE
  800 FORMAT(3(F10.7),14X,I1,4X,I1,4X,I4,1X,I1,4X,I1,4X,I1)
    9 READ(60,10)VULCAN,KILL
   10 FORMAT(F10.5,10X,I1)
      IFRST(1)=1
      IFRST(2)=1+   LAST
      IFRST(3)=1+ 2*LAST
      IFRST(4)=1+ 3*LAST
      IFRST(5)=1+ 4*LAST
      IFRST(6)=1+5*LAST
      IFRST(7)=1+6*LAST
      IFRST(8)=1+7*LAST
      IFRST(9)=1+8*LAST
      ILST(1)=  LAST
      ILST(2)=2*LAST
      ILST(3)=3*LAST
      ILST(4)=4*LAST
      ILST(5)=5*LAST
      ILST(6)=6*LAST
      ILST(7)=7*LAST
      ILST(8)=8*LAST
      ILST(9)=9*LAST
      XMIN=0.
      XMAX1=EPSILON*LAST
      XTICK=XMAX1/10.
      XMAX=(NCURVES-1)*XTICK/2.+XMAX1
      XSIZE=XMAX/XTICK
      KC=KC+1
      ORSHIFTX=XTICK/2.
      YMIN=0.
      YMAX1=1.
      YTICK=YMAX1/10.
      ORSHIFTY=YTICK/4.
      YMAX=(NCURVES-1)*ORSHIFTY+YMAX1
      IF (JELLO.EQ.1) GO TO 2
    1 PRINT 6
      GO TO 3
    2 PRINT 7
    3 PRINT 12,VULCAN,FERMI,EZERO,XMAX1
      PRINT 8
    6 FORMAT (1X,*PHASE SHIFTS IN RADIANS FOR A QUANTUM MECHANICAL PARTI
     1CLE SCATTERING OFF A 3-D SQUARE WELL.*/)
    7 FORMAT (1X,*PHASE SHIFTS IN DEGREES FOR A QUANTUM MECHANICAL PARTI
     1CLE SCATTERING OFF A 3-D SQUARE WELL.*/)
    8 FORMAT(5X,*ENERGY(I)*4X*DELTA(0)*4X*DELTA(1)*4X*DELTA(2)*4X*DELTA(
     13)*4X*DELTA(4)*4X*    K    *4X*  ERGO  *4X*OLYMPUS1*4X*HADES1*/)
   12 FORMAT(1X,*WELL DEPTH= *F10.5,*  WELL WIDTH= *F10.7,*  EZERO= *
     1 F10.7,* ELAST= *F10.7/)
      K=MAX+1
      ENERGY(1)=EZERO
      DO 91 I=1,LAST
      ERGO(I)=ENERGY(I)*(FERMI)**2
   14 ENERGY(I+1)=EPSILON+ENERGY(I)
   15 CYCLOPS(I)=SQRT(ENERGY(I))
   20 CENTAUR(I)=SQRT(ENERGY(I)+VULCAN)
   25 ZEUS(I)=CYCLOPS(I)*FERMI
      ZEUSS=ZEUS(I)
   27 IF(ZEUSS.EQ.0) 78,30
   30 RHO(I)=CENTAUR(I)*FERMI
      RHOO=RHO(I)
   34 CALL SPHJN(RHOO,K,B)
      CALL SPHYN(RHOO,K,N)
      DB(1)=-B(2)
      DN(1)=-N(2)
      DO 35 JI=2,MAX
      DB(JI)=(JI*B(JI-1)-(JI+1)*B(JI+1))/(2*JI+1)
      DN(JI)=(JI*N(JI-1)-(JI+1)*N(JI+1))/(2*JI+1)
   35 CONTINUE
      CALL SPHJN(ZEUSS,K,BZ)
      CALL SPHYN(ZEUSS,K,NZ)
      DBZ(1)=-BZ(2)
      DNZ(1)=-NZ(2)
   37 DO 38 KI=2,MAX
      DBZ(KI)=(KI*BZ(KI-1)-(KI+1)*BZ(KI+1))/(2*KI+1)
      DNZ(KI)=(KI*NZ(KI-1)-(KI+1)*NZ(KI+1))/(2*KI+1)
   38 CONTINUE
   40 DO 76 LI=1,MAX
   44 GAMMA(LI)=CENTAUR(I) *DB(LI)/B(LI)
   45 CASTOR(LI)=CYCLOPS(I)*DBZ(LI)
   50 POLLUX(LI)=GAMMA(LI)*BZ(LI)
   55 HERCULES(LI)=CYCLOPS(I)*DNZ(LI)
   60 AJAX(LI)=GAMMA(LI)*NZ(LI)
   68 OLYMPUS(LI)=CASTOR(LI)-POLLUX(LI)
   70 HADES(LI)=HERCULES(LI)-AJAX(LI)
   75 W(LI)   =ATAN2(OLYMPUS(LI),HADES(LI))
      IF(W(LI).LT.0) W(LI)=W(LI)+3.14159
      IF(JELLO.EQ.0) GO TO 76
      W(LI)=W(LI)*57.29578
   76 CONTINUE
      IF(JELLO.EQ.1)GO TO 80
      W(LI)=W(LI)/3.14159
   77 GO TO 80
C  THIS SETS DELTA = ZERO FOR E=0
   78 DO 79 LI=1,MAX
      W(LI)=0.
   79 CONTINUE
      OLYMPUS(1)=0.
      HADES(1)=0.
   80 PRINT 85,ENERGY(I),W(1),W(2),W(3),W(4),W(5),ZEUS(I),ERGU(I),OL
     1YMPUS(1),HADES(1)
   85 FORMAT (10(2X,F11.7))
      DELTA(I)=W(1)
      DELTA1(I)=W(2)
      DELTA2(I)=W(3)
      DELTA3(I)=W(4)
      DELTA4(I)=W(5)
   91 CONTINUE
   86 PRINT 87
   87 FORMAT (///)
C   SUBROUTINE MINIMAX AUTOMATICALLY FINDS YMIN AND YMAX FOR LAYOUT.
C   YMIN=DELTA(MIN), YMAX=DELTA(MAX), XMIN=EZERO,XMAX= EPSILON*LAST.
        IF(NOPLOT.EQ.1) GO TO 95
        IF (NOSCALE.EQ.1)GO TO 99
        CALL MINIMAX(DELTA,YMIN,YMAX,LAST)
C THE PLOT ROUTINE, LAYOUT, IS DUE TO STEVE BRENNER,CODE  )NRL
C XSIZE=PHYSICAL LENGTH OF PLOT IN INCHES.  XMIN =MIN. VALUE OF ABSCISSA
C  XMAX=MAX VALUE OF ABSCISSA.  (XMAX-XMIN)/XSIZE DETS VALUE OF TIC
C  MARKS.  SIMILARLY Y REFERS TO ORDINATE, YSIZE IS FIXED AT 10 INCHES.
C  IN THE CALL FOR SUBROUTINE LAYOUT, THE PARAMETERS IN THE CALL STATE-
C  MENT ARE RESPECTIVELY ABSCISSA VARIABLE, ORDINATE VARIABLE, NUMBER OF
C  DATA POINTS(=LAST),NO. OF HOLLERITH CHARACTERS FOLLOWED BY ABSCISSA
C  IDENT.,ORDINATE IDENT, E.G., 8HDELTA-0.  (8 CHARACTERS MAX), FOLLOWED
C  BY THE ACTUAL NUMBER OF CHARACTERS USED IN THE ABSS.,ORD. NAMES.
C  FINALLY THE TOTAL NUMBER OF PLOTS DESIRED SUPERIMPOSED ON THE SAME
C  GRAPH.  IF ONE PLOT PER GRAPH, USE 1.  IF N, USE N.  HOWEVER, IF N
C  PLOTS ARE SUPERIMPOSED, THE THE DIMENSION STATEMENTS FOR THE ORD,ABS
C  AND ALSO LAST MUST BE SCALED BY N.  IF SELF-SCALING IS DESIRED, SET
C  SUBROUTINE MINIMAX AUTOMATICALLY FINDS YMIN AND YMAX FOR LAYOUT.
C  YMIN=DELTA(MIN), YMAX=DELTA(MAX), XMIN=EZERO,XMAX= EPSILON*LAST.
C  THE PLOT ROUTINE WILL SCALE  THE PLOTTED VALUES BY POWERS OF TEN.
C IF SELF-SCALING NOT USED8 THEN YMIN AND YMAX VALUES MUST BE SUPPLIED.
C IF DESIRED, DIFFERENT ORDER PHASE SHIFTS MAY BE PLOTTED ON THE SAME
C  GRAPH.  USE NOSCALE TO BYPASS, PUT A 1 IN COL 70.  SUPPLY YMIN,YMAX.
   99 CONTINUE
        NLAST=NCURVES*LAST
        J=LAST
        ORSHIFTX=ORSHIFTX*KC
        ORSHIFTY=ORSHIFTY*KC
        JAZZ=KC*J
        DO 90 M=1,J
        PHASE(M+JAZZ)=DELTA(M)+ORSHIFTY
      FRGON(M+JAZZ)=ERGO(M)+ORSHIFTX
   90 CONTINUF
      PRINT 100
  100 FORMAT(///)
   95 IF(KILL.EQ.0) GO TO 9
      IF(NOPLOT.EQ.1)GO TO 199
      CALL LAYOUT(ERGON,PHASE,NLAST,6HE*A**2,7H-DELTA-,6,7,NCURVES)
      CALL STOPPLOT
  199 END
      SUBROUTINE MINIMAX(ARRAY,YMIN,YMAX,LAST)
      DIMENSION ARRAY(1000)
C   THIS SUBROUTINF FINDS BOTH THE LARGEST AND SMALLEST NUMBERS IN ARRAY.
C  LARGFST NUMBER
      J=1
      JJ=1+J
   10 DO 30 K=JJ,LAST
      IF(ARRAY(J).GF.ARRAY(K))30,20
   20 J=K
      IF(K.EQ.LAST) GO TO 40
       GO TO 50
   30 CONTINUF
   40 YMAX=ARRAY(J)
      PRINT 45,JJ
   45 FORMAT (1X,*JJ=*I3)
      GO TO 60
   50 JJ=J+1
      GO TO 10
C  SMALLFST NUMBFR
   60 L=1
      LL=1+L
  100 DO 300 M=LL,LAST
      IF(ARRAY(L).LF.ARRAY(M))300,200
  200 L=M
      IF(M.EQ.LAST)GO TO 400
      GO TO 500
  300 CONTINUF
  400 YMIN=ARRAY(L)
      PRINT 450,LL
  450 FORMAT (1X,*LL=*I3)
      GO TO 600
  500 LL=L+1
      GO TO 100
  600 RETURN
      END
      SUBROUTINF SPHJN(X,N,ARR)
C3 NPL BESN       BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS
      DIMENSION ARR(1)
      RX=1.0E+307
      NF=1
      NN=N+1
      IF(X.EQ.0.0) GO TO 4
      IF(X.LE.0.5)GO TO 6
C     FOR ORDERS LFSS THAN ARGUMFNT, AND ARGUMENT GREATER THAN 0.5
      ARR(1)=SIN(X)/X
      ARR(2)=ARR(1)/X-COS(X)/X
      NM=X-2
      IF(N.LT.X)NM=N-1
      IF(NM.LT.0)NM=0
      DO 16 I=1,NM
      FN=I+I
      ARR(I+2)=(FN+1.0)*ARR(I+1)/X-ARR(I)
C     CHECK FOR UNDERFLOW
      IF(ARR(I+2).NE.0.0)GO TO 16
      ARR(I+1)=ARR(I+1)*RX
      ARR(I+2)=(FN+1.0)*ARR(I+1)/X-ARR(I)*RX
      PRINT 13, X,I,N
   16 CONTINUE
      IF(N.LT.X)RETURN
      HOLD=ARR(NM+2)
C     FOR ORDERS GREATER THAN OR EQUAL TO ARGUMENT, AND ARGUMENT
C     GREATER THAN 0.5
   15 ALPHA=1.0
      M=-2-NM
      NOTE=0
      I=N+47
      IF(X/N.GT.0.9)I=I+X/10.0
      ARR(I+2)=ALPHA
      ARR(I+3)=0.0
      I=-I-1
      DO 2 KK=I,M
      K=-KK
      NOO=K+1
      TKP1=K+K+1
      ARR(K)=TKP1*ARR(K+1)/X-ARR(K+2)
      IF EXPONENT FAULT 18,2
   18 TEMPY=ARR(K)=ARR(K+1)/RX*TKP1/X-ARR(K+2)/RX
      ARR(K)=ARR(K)*RX
      IF EXPONENT FAULT 24,2
   24 ARR(K)=TEMPY
      ARR(K+1)=ARR(K+1)/RX
      PRINT 13,X,N,NOO
    2 CONTINUE
   12 C=HOLD/ARR(K)
      IF(C.NE.0.0)GO TO 14
      HOLD=HOLD*RX
      NOTE=NOTE+1
      GO TO 12
   14 IF(NOTE.EQ.0)GO TO 17
      MULT=307*NOTE
      NUL=L-1
      PRINT 21,X,N,NUL,MULT
   21 FORMAT (3X*FOR ARGUMENT OF *E17.10* AND ORDERS OF *I4* TO *I4,
     1          *SPHJN EXPONENT WAS INCREASED BY *I6)
   17 DO 3 I=K,NN
      ARR(I)=ARR(I)*C
    3 CONTINUE
      RETURN
    4 ARR(1)=1.0
      DO 5 L=2,NN
      ARR(L)=0.0
    5 CONTINUE
      RETURN
C     SERIES FOR ARGUMENT LESS THAN OR EQUAL TO 0.5
    6 EPS=5.0E-11
      CH=CS=DN=DIV=1.0
      DO 10 L=1,NN
      ITEXP=NEXPA=MEXPA=LEXPA=0
      NO=L-1
      IF(L.EQ.1) GO TO 7
      CS=CS*X/DN
C     CHECK FOR UNDERFLOW
      IF(CS.NE.0.0) GO TO 7
C     SCALE FOR THIS N AND ALL FOLLOWING N
      CS=CH*RX*X/DN
      NEXPA=NEXPA+1
    7 DN=DN+2.0
      CH=CS
      XKK=0.0
      A=CH2=FORTT=1.0/DIV
      TN=2.0*NO
      FN=-0.5*X*X
    8 XKK=XKK+1.0
      FD=XKK*(TN+XKK+XKK+1.0)
      A=A*FN/FD
C     CHECK FOR UNDERFLOW
      IF(A.NE.0.0)GO TO 25
C     SCALE FOR THIS N AND ALL FOLLOWING N
      A=CH2*RX*FN/FD
      DIV=DIV/RX
      MFXPA=MEXPA+1
      FORTT=FORTT*RX
   25 CH2=A
      TT=A+FORTT
      IF(TT.EQ.0.0.OR.ABS(A/TT).LT.EPS) GO TO 9
      FORTT=TT
      GO TO 8
    9 ARR(L)=CS*TT
C     CHFCK FOR UNDFRFLOW
      IF(ARR(L).NF.0.0)GO TO 31
C     SCALE FOR THIS N AND ALL FOLLOWING N
      ARR(L)=CS*RX*TT
      DIV=DIV/RX
      LEXPA=LEXPA+1
   31 ITEXP=NFXPA+MEXPA+LEXPA
      IF(ITEXP.FQ.0)GO TO 10
      IF(ITEXP.FQ.1)GO TO 29
      PRINT 30, NEXPA,MEXPA,LEXPA
   30 FORMAT(3X,* SCALE FRROR *3(3X,I10))
   29 PRINT 13, X,NO,N
   13 FORMAT(3X,*FOR ARGUMENT OF *,E17.10,* AND ORDERS OF *,I4,* TO *,
     1 I4,* SPHJN EXPONFNT WAS INCREASED BY 307*)
   10 CONTINUF
      FND
      SUBRCUTINF SPHYN(X,N,ARR)
      DIMENSION ARR(1)
      RX=1.0E+307
      IF(X.EQ.0.)5,3
    5 PRINT 4
    4 FORMAT(3X,*DO NOT USE ZERO ARGUMENT FOR SPHYN*)
      RETURN
C     FIND Y0 AND Y1
    3 ARR(1)=-COS(X)/X
      ARR(2)=ARR(1)/X-SIN(X)/X
      NM1=N-1
      DO 2 K=1,NM1
      TKP1=K+K+1
      ARR(K+2)=TKP1*ARR(K+1)/X-ARR(K)
```

```
C       CHECK FOR OVERFLOW
        IF EXPONENT FAULT 8,2
      8 ARR(K+1)=ARR(K+1)/RX
        ARR(K+2)=TKP1*ARR(K+1)/X-ARR(K)/RX
        PRINT 7,X,K,N
      7 FORMAT(3X,*FOR ARGUMENT OF *,F17.10,* AND ORDERS OF *,I4,* TO *,
     1 I4,* SPHYN EXPONENT WAS DECREASED BY 307*)
      2 CONTINUE
        END
        SUBROUTINE LAYOUT (XBFFR,YBFFR,NN,XBCD,YBCD,NNX,NNY,NUMBER)
        COMMON/LABEL/IL,ILX(9),ILY(9),ILH(9),ILR(9),ILC(9),ILABLE(9,9)
        COMMON /RANGE/ XMAX,XMIN,YMAX,YMIN,IFRST(9),ILST(9),XSIZE
        REAL ILX,ILY,ILH,ILR
        DIMENSION XBCD(5),YBCD(5)
        DIMENSION XBFFR(NN),YBFFR(NN),PLTARRAY(254)
        DATA (ISTRT=1)
C                 INITIALIZATION
        N=NN
        NX=NNX
        NY=NNY
        IF (XSIZE.EQ.0.0) XSIZE=10.
        YSIZE=10.
        NSVE=N
        N=XABSF (N)
        IF (NUMBER.EQ.0) NUMBER=1
        IF (ISTRT.EQ.2) GO TO 2
        CALL PLOTS(PLTARRAY,254,13)
        ISTRT=2
      2 CONTINUE
C
C                 LABEL X AXIS
C

        DX= (XMAX-XMIN)/XSIZE
        IF (DX.NE.0) GO TO 3
        CALL SCALE (XBFFR,N,XSIZE,XMIN,DX,1)
        GO TO 7
      3 CONTINUE
        DO 6 I=1,N
C       PRINT 5555,XBFFR(I),YBFFR(I),I
   5555 FORMAT (2F20.10,I10)
      6 XBFFR(I)= (XBFFR(I)-XMIN)/DX
      7 CONTINUE


C       CALL SCALE (XBFFR,N,XSIZE,XMIN,DX,1)
        ZERO=XSIZE-ZERO
        IF (NSVE) 5,10,10
      5 XMIN=XSIZE*DX-ABSF(XMIN)
        DX=-DX
     10 CONTINUE
        ZERO=(0.0-XMIN)/DX
        PRINT 4,DX,XMIN,ZERO
      4 FORMAT (* DX=*,3F20.5)
        CALL AXIS (0.0,0.0,XBCD,NX,XSIZE,0.0,XMIN,DX)
        PRINT 4,DX,XMIN,ZERO
        NX=-NX
        CALL AXIS (0.0,YSIZE,XBCD,NX,XSIZE,0.0,XMIN,DX)
C
```

```
      PRINT 4,DX,XMIN,ZERO
C             LABEL Y AXIS
C


      DY= (YMAX-YMIN)/YSIZE
      IF (DY.NE.0.0) GO TO 14
      CALL SCALE (YBFFR,N,YSIZE,YMIN,DY,1)
      GO TO 15
   14 CONTINUE
      DO 16 I=1,N
   16 YBFFR(I)=(YBFFR(I)-YMIN)/DY
   15 CONTINUE


      PRINT 11,DY,YMIN
   11 FORMAT (* DY=*,3F20.4)
      CALL AXIS (0.0,0.0,YBCD,NY,YSIZE,90.,YMIN,DY)
      NY=-NY
      PRINT 11,DY,YMIN
      CALL AXIS (XSIZE,0.0,YBCD,NY,YSIZE,90.,YMIN,DY)
      PRINT 11,DY,YMIN
C
C             PLOT CURVE
      NUMBER1=NUMBER
      IF (NUMBER.LT.0) NUMBER=-NUMBER
      ND=(N/NUMBER)-1
      N2=0
      IF (NUMBER.NE.1) GO TO 8
      IFRST(1)=1
      ILST(1)=N
    8 CONTINUE
      DO 150 J=1,NUMBER
      IF (NUMBER1.LT.0) GO TO 148
      N1=IFRST(J)
      N2=ILST(J)
      GO TO 149
  148 CONTINUE
      N1=N2+1
      N2=N1+ND
  149 CONTINUE
      IPN=3
C
C     CALL PLOT (XBFFR(1),YBFFR(1),3)
C
      DO 100 I=N1,N2
C     PRINT 5555,XBFFR(I),YBFFR(I),I
      XBF= XBFFR(I)
      YBF= YBFFR(I)
      IF (NSVF.LT.0) XBFFR(I)=XSIZE-XBFFR(I)
      IF (XBF.LT.0.0.OR.XBF.GT.XSIZE) GO TO 100
      IF (YBF.LT.0.0.OR.YBF.GT.YSIZE) GO TO 100
      CALL PLOT(XBF,YBF,IPN)
      IPN=2
  100 CONTINUE
  150 CONTINUE
      IF (IL.EQ.0) GO TO 255
      ICT=0
  252 ICT=ICT+1
```

```
      IF (ICT.GT.IL) GO TO 255
      CALL SYMBOL(ILX(ICT),ILY(ICT),ILH(ICT),ILABLE(1,ICT),ILR(ICT)
     1,ILC(ICT))
      GO TO 252
  255 CONTINUF
      CALL PLOT (XSIZE+3.,0.0,-3)
      PRINT 250
  250 FORMAT (* PLOT DONE*)
C
C

C         RESTORE THE DATA
      DX=ABS(DX)
      DY=ABS(DY)
C
      DO 400 I=1,N
      XBFFR(I)=(DX*XBFFR(I))+XMIN
      YBFFR(I)=(DY*YBFFR(I))+YMIN
  400 CONTINUE
C
      XMAX=0.0
      XMIN=0.0
      YMAX=0.0
      YMIN=0.0
C
      RETURN
      END
      SUBROUTINF SCALE (X,N,S,YMIN,DY,K)
      DIMENSION X(2)
      YMAX=X(1)
      YMIN=YMAX
      NP=N*K
      DO 10 I=1,NP,K
      IF(YMAX-X(I))5,6,6
    5 YMAX=X(I)
    6 IF(X(I)-YMIN) 7,10,10
    7 YMIN=X(I)
   10 CONTINUE
      DY=(YMAX-YMIN)/ (S)
      DO 20 I=1,N
   20 X(I)=(X(I)-YMIN)/DY
      PRINT 25,YMIN,YMAX,DY,X(1),X(N)
   25 FORMAT(5F25.5)
      RETURN
      END
      SUBROUTINE AXIS (X,Y,BCD,NC,SIZE,THETA,YMIN,DY)
      X14=.07
      SIGN=1.0
      IF (NC) 1,2,2
    1 SIGN=-1.0
    2 NAC=XABSF(NC)
      TH=THETA*0.017453294
      N=SIZE+.50
      CTH=COSF(TH)
      STH=SIN(TH)
      TN=N
      XB=X
      YB=Y
      XA=X-.1*SIGN*STH
      YA=Y+.1*SIGN*CTH
C     IF (THETA-90.)10,12,12
C  12 XA=-XA
   10 CALL PLOT (XA,YA,3)
C
C DO LOOP DRAWS AXIS AND MAKES TIC MARKS
C
      DO 20 I=1,N
      CALL PLOT (XB,YB,2)
      XC=XB+CTH
      YC=YB+STH
      CALL PLOT (XC,YC,2)
      XA=XA+CTH
      YA=YA+STH
      CALL PLOT (XA,YA,2)
      XB=XC
   20 YB=YC
      CHAR=ABSF (YMIN)
      ABSV= ABSF (YMIN+DY)
      IF (ABSV-CHAR) 5,6,6
    5 ABSV=CHAR
    6 FXP=0.0
   89 NT=ABSV+0.1
      CHAR=NT
      IF (CHAR-ABSV) 90,92,92
   90 FXP=FXP-1.0
      ABSV=ABSV*10.
      GO TO 89
   91 FXP=FXP+1.0
   92 ABSV=ABSV/10.0
      NT=ABSV
      CHAR=NT
C     IF (CHAR-ABSV)93,91,91
      IF (ABSV.GT.1) GO TO 91
   93 ADY=DY*10.0**(-FXP)
      ABSV=YMIN*10.0**(-EXP)+TN*ADY
      XA=XB-(.20*SIGN-0.05)*STH-.25*CTH
      YA=YB+(.20*SIGN-.05)*CTH-.25*STH
      N=N+1
C     IF (THETA-90.) 31,32,32
C  32 XA=-XA+.1
C
C DO LOOP PLOTS NUMBERS ALONG AXIS
C
   31 DO 30 I=1,N
      CALL NUMBER (XA,YA,.07,ABSV,THETA,4HF7.3)
      ABSV=ABSV-ADY
C     IF (ABSV.LT.0.AND.ABSV.GT.-1) 55,56
C  55 ABSV=0
   56 XA=XA-CTH
   30 YA=YA-STH
      TNC=NAC+7
C
C CALCULATES CENTER POSITION FOR LABEL
      XA=X+(SIZE/2.0-.06*TNC)*CTH-(-.07+SIGN*.36)*STH
      YA=Y+(SIZE/2.0-.06*TNC)*STH+(-.07+SIGN*.36)*CTH
C     IF (THETA-90.) 41,42,42
C  42 XA=-XA+.1
   41 CALL SYMBOL (XA,YA,X14,BCD,THETA,NAC)
```

```
C
C PRINTS LABEL
      XA=XA+((TNC-6.0)*0.12)*CTH
      YA=YA+((TNC-6.0)*0.12)*STH
      IF (EXP) 35,50,35
   35 CALL SYMBOL (XA,YA,X14,8H(X10    ),THETA,8)
      XA=XA+.48*CTH-.07*STH
      YA=YA+.48*STH+.04*CTH
      IF (EXP) 40,50,40
C
C   PRINTS SCALF FACTOR
   40 CALL NUMRFR (XA,YA,X14,FXP,THETA,4HF3.0)
   50 CONTINUF
      CALL PLOT (XA,YA,3)
      RETURN
       END
''GG
''GG
''GG
```